UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

Sistema de búsqueda de respuestas sobre DBpedia

Autor: Eduardo Ruiz de Pascual Núñez

Tutor: Iván Cantador Gutiérrez

MAYO 2016

Sistema de búsqueda de respuestas sobre DBpedia

AUTOR: Eduardo Ruiz de Pascual Núñez
TUTOR: Iván Cantador Gutiérrez

Grupo de Recuperación de Información
Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Mayo de 2016

Resumen (castellano)

La búsqueda de respuestas a preguntas concretas en los contenidos de la Web es en muchas ocasiones una tarea difícil y costosa. Por una parte, esto es debido a la ingente y constantemente creciente cantidad de información disponible. Por otra parte, porque los motores de búsqueda actuales están basados en la coincidencia de palabras clave y recuperan aquellos documentos en los que se dan ocurrencias de las palabras usadas en las consultas, sin analizar, comprender y explotar la semántica (significados y relaciones) subyacente, tanto en consultas como en documentos.

Los Sistemas de Búsqueda de Respuestas –del inglés Question Answering (QA) systems– pretenden dar solución al problema anterior, permitiendo al usuario realizar consultas en lenguaje natural –en vez de por medio de palabras clave–, y dando como resultados respuestas concretas ya procesadas y verificadas, presentadas también en lenguaje natural en vez de en un listado de documentos.

Question Answering es un problema de investigación complejo abierto, no resuelto satisfactoriamente, especialmente cuando se intentan tratar preguntas no restringidas sintácticamente y sobre múltiples dominios.

Abordando esas limitaciones, en este Trabajo de Fin de Grado se plantea desarrollar un prototipo de QA que haga uso de herramientas de Procesado de Lenguaje Natural avanzadas para el procesamiento y comprensión de preguntas abiertas y que acceda de manera flexible a bases de conocimientos de la Web Semántica –en particular a DBpedia, la versión estructurada de Wikipedia–, para la obtención de las respuestas correspondientes a tales preguntas.

Palabras clave (castellano)

Sistema de Búsqueda de Respuestas, Procesamiento del Lenguaje Natural, Web Semántica, Linked Data, DBpedia.

Abstract

Searching for answers to specific questions on the Web is often a difficult and time-consuming task. This is due, on the one hand, to the huge and ever growing amount of available information and the fact that current search engines merely return long lists of documents potentially relevant to the queries made. On the other hand, it is because such systems are based on keyword matching and do not analyze and understand the underlying semantics (concepts and relations) in both queries and documents.

Question Answering Systems intend to address that problem, allowing the user to ask questions in natural language –instead of keyword-based queries–, and retrieving specific –and already processed and verified– answers in natural language, rather than lists of documents.

Question Answering is an open research problem, which has not been solved satisfactory in many cases, especially when dealing with open, non restricted questions on multiple domains.

Addressing these last limitations, this Bachelor Thesis aims to develop a prototype Question Answering System, using advanced Natural Language Processing tools to process and understand open domain questions, and exploiting in a general and flexible way knowledge bases from the Semantic Web –DBpedia, the structured version of Wikipedia, in particular– to automatically obtain the answers to the input questions.

Keywords

Question Answering Systems, Natural Language Processing, Semantic Web, Linked Data, DBpedia.

Glosario

API ***Application Programming Interface***

Conjunto de procedimientos que ofrece una biblioteca para poder acceder a los recursos de un sistema.

HTML ***HyperText Markup Lenguage***

Lenguaje estándar del desarrollo Web, basado en etiquetas de marcado.

HTTP ***HyperText Transfer Protocol***

Protocolo de comunicación que permite las transferencias de hipertexto en la Web.

PLN (o NLP) ***Procesamiento de Lenguaje Natural (Natural Language Processing)***

Estudio centrado en el procesamiento del lenguaje natural para extraer conocimientos semánticos de la oración analizada.

RDF ***Resource Description Framework***

Familia de especificaciones de la *World Wide Web Consortium* originalmente diseñado como un modelo de datos para metadatos, que es usado como un método general para la descripción conceptual o modelado de la información que se implementa en los recursos Web. Está basado en XML.

SPARQL ***SPARQL Protocol and RDF Query Language***

Lenguaje formal estandarizado para la consulta de grafos (bases de datos) RDF inspirado en SQL.

QA ***Question Answering***

Disciplina de Ciencias de la Computación que confluye de la intersección de los campos de la Recuperación de Información y del Procesamiento del Lenguaje Natural, cuyo objetivo final es el de desarrollar sistemas informáticos que automáticamente dan respuestas a preguntas planteadas por personas en lenguaje natural.

SQL ***Structured Query Language***

Lenguaje estandarizado para la consulta y definición de bases de datos relacionales.

URI ***Uniform Resource Identifier***

Cadena de caracteres que identifica un recurso de manera unívoca. Permite la interacción con representaciones del recurso en una red, típicamente la Web, usando un protocolo específico, e.g., HTTP.

XML ***Extensible Markup Language***

Lenguaje de marcado que permite definir la gramática y estructura de lenguajes de representación de información específicos.

Agradecimientos

Muchas son las personas que, directa o indirectamente, han contribuido a que mi periplo universitario haya sido más llevadero y sería injusto no acordarme de ellos ahora que estoy a punto de acabar; me gustaría dar las gracias, en primer lugar, a todos los profesores que, con su dedicación diaria, consiguieron despertar en mi un gran interés por la informática en todos sus ámbitos y que han contribuido, de manera decisiva, en mi formación. Del mismo modo, me gustaría acordarme de todos los compañeros con los que empecé este viaje y a los cuales hoy tengo la suerte de llamar amigos: Miguel, Álvaro, Alberto, Sorasu, Víctor, Borja, Manu y tantos otros que han compartido conmigo las frustraciones y agobios de esta o aquella entrega y sin los cuales hoy no estaría presentando esta memoria. No podría dejar fuera de la lista a Roberto Veral, amigo con el que he tenido la suerte de compartir estos 4 años y muchas historias que trascienden a la carrera.

En la realización de este trabajo ha sido indispensable la ayuda, colaboración y paciencia de Iván Cantador, tutor del mismo, que ha hecho posible que algo que veía irrealizable hace unos meses se haya convertido en realidad.

No querría olvidarme tampoco de las personas ajenas a la carrera que han estado ahí desde mucho antes de mi primera clase en la Autónoma; gracias a Sergio, a Mónica, a Pepón, a Diego y tantos otros. Gracias a Miriam, por animarme en los días malos y celebrar conmigo los buenos.

Por último, gracias a mis padres, por su apoyo incondicional y su esfuerzo por convertirme en la persona que soy. Mis éxitos son tan suyos como míos.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

1 Introducción

1.1 Motivación

La búsqueda de respuestas a preguntas concretas en los contenidos de la Web es, en muchas ocasiones, una tarea difícil y costosa. Por una parte, esto es debido a la ingente y constantemente creciente cantidad de información disponible y al hecho de que los motores de búsqueda actuales se limitan a devolver grandes listas de documentos, que potencialmente son relevantes a las consultas realizadas [1]. Por otra parte, porque dichos buscadores están basados en coincidencia de palabras clave y recuperan aquellos documentos en los que se dan ocurrencias de las palabras usadas en las consultas, sin analizar, comprender y explotar la semántica (significados y relaciones) subyacente, tanto en consultas como en documentos. Así, por ejemplo, cuando se lanza la consulta "películas de 1997 en las que participó Leonardo DiCaprio" a un buscador Web, éste devolvería documentos en los que se mencione a Leonardo DiCaprio; algunos sobre el propio actor, otros sobre sus películas y otros que podrían no tener una relevancia o relación clara/principal con el actor, y en cualquier caso sin establecer una comprensión clara de la pregunta y los documentos, para poder devolver la respuesta concreta ("Titanic", de James Cameron) o al menos recuperar sólo los documentos que la tuviesen.

Los **Sistemas de Búsqueda de Respuestas** –del inglés Question Answering (QA) systems– pretenden dar solución al problema anterior, permitiendo realizar consultas en lenguaje natural y dando como resultados respuestas concretas ya procesadas y verificadas, presentadas también en lenguaje natural en vez de en un listado de documentos.

Question Answering es así una disciplina de Ciencias de la Computación que confluye de la intersección de los campos de la Recuperación de Información y del Procesamiento del Lenguaje Natural (PLN), cuyo objetivo final es desarrollo de sistemas informáticos que automáticamente dan respuestas a preguntas planteadas por personas en lenguaje natural –usando un determinado idioma. En este proceso, los sistemas de Búsqueda de Respuestas, pueden llevar a cabo la obtención de respuestas bien realizando consultas formales sobre una base de datos estructurada con conocimiento en uno o varios dominios, bien recuperando y procesando información existente en una colección de documentos no estructurados, como las páginas Web de la enciclopedia en línea Wikipedia. En ambos casos, el sistema ha de tratar con una amplia variedad de tipos de preguntas, incluyendo preguntas sobre datos concretos (¿Qué…?, ¿Cuál/es..?), fechas (¿Cuándo…?), lugares (¿Dónde…?), maneras o modos (¿Cómo…?), cantidades (¿Cuántos/as…?) y razones (¿Por qué..?), entre otros.

En la Figura 1 se muestran los principales componentes de un sistema básico de Question Answering, incluyendo el análisis de la pregunta introducida por el usuario, la composición de la consulta (formal) que pueda utilizarse para acceder a la base de conocimiento empleada y finalmente, la obtención y procesamiento de las respuestas en el lenguaje natural de la pregunta realizada. Atendiendo a este esquema, un sistema de QA se puede considerar como un sistema que aborda dos problemas complejos: 1) el procesamiento del lenguaje natural de la pregunta introducida, con el fin de extraer su semántica subyacente y construir una consulta en un lenguaje formal a ejecutar sobre una base de conocimiento; y 2) la recuperación, validación y procesamiento de respuestas desde la base de conocimiento a partir de la consulta formal generada.

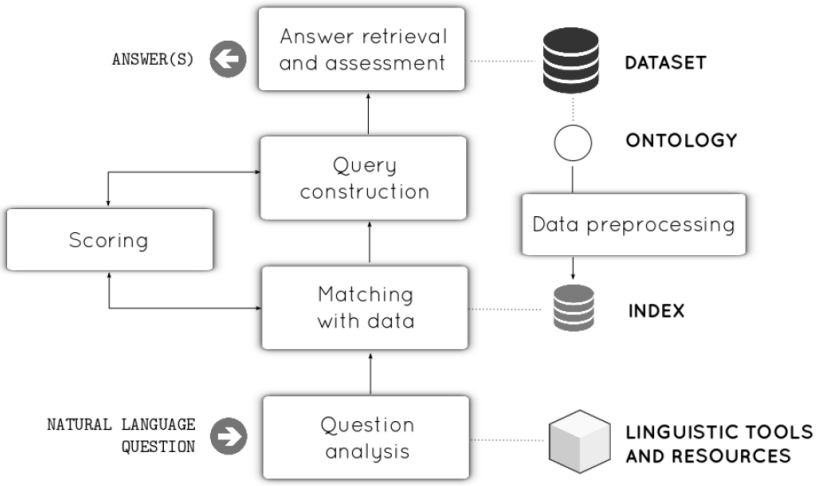

Figura 1. Estructura básica de un sistema de Question Answering [12].

En función del alcance de cada una las componentes anteriores, han ido apareciendo distintas aplicaciones, en general enfocadas a simplificar la tarea de búsqueda de respuestas [1]. Por ejemplo, en algunos servicios de atención al cliente, el usuario no introducirá su consulta en lenguaje natural, sino que elegirá entre un conjunto de preguntas predefinidas planteadas por el sistema para determinar el motivo de su llamada y recibir la respuesta o atención oportuna. Este tipo de aplicaciones no sólo tienen limitadas las preguntas a tratar, sino que también dichas preguntas suelen abordar muy pocos temas o dominios. En otros sistemas más complejos, a la vez que menos eficaces, se acepta una amplia gama de preguntas ajustadas a patrones predefinidos, cubriendo varios temas y dominios. Así, en la literatura se distinguen dos tipos de sistemas de QA en función del dominio o dominios abordados:

1. **Sistemas de Búsqueda de Respuestas de dominio cerrado** (del inglés, closed-domain QA systems), que únicamente responden preguntas de un dominio determinado (como medicina, deportes, historia…). De esta forma, la pérdida de potencia del sistema a la hora de responder preguntas de otro ámbito se ve compensada por la profundidad que puede alcanzar en el dominio en el que se enfoca. Otra alternativa de este tipo de sistema, supone la limitación, no en el ámbito conocido, sino en el tipo de preguntas respondido (de tipo quién, cómo, cuándo…), haciendo así más simple el procesado del lenguaje natural de entrada, al reducir las posibilidades sintácticas de éstas.
2. **Sistemas de Búsqueda de Respuestas de dominio abierto** (del inglés, open-domain QA systems), que dan respuesta a preguntas de cualquier ámbito, apoyadas en ontologías y bases de conocimiento de gran tamaño.

En este trabajo se pretende ofrecer una solución de Sistema de Búsqueda de Respuestas de dominio abierto, que no presente limitación en la temática de las preguntas introducidas.

El acceso y recuperación de información de la base de conocimiento utilizada dependerá del tipo de fuente empleada. Si se utiliza una base de datos relacional, será suficiente con utilizar un lenguaje de consulta, como SQL; si se intenta realizar la búsqueda sobre toda la Web, la componente de recuperación de información deberá ser mucho más compleja [3].

Teniendo en cuenta lo anterior, se puede considerar como una aproximación intermedia el uso de la conocida como Web Semántica [4], que se refiere a una implementación de la Web que añade semántica a sus contenidos. Dicha semántica vendría dada por anotaciones acerca del significado de los conceptos o entidades (objetos, personas, organizaciones, eventos, etc.) mencionados en los páginas Web, así como las atributos y relaciones de/entre tales conceptos y entidades. Además, estos conceptos, entidades, atributos y relaciones estarían definidos formalmente por medio de ontologías[1] y bases de conocimiento estructuradas, accesibles mediantes lenguajes de consultas similares a los usados en bases de datos relacionales [5].

De acuerdo a este principio, la información de la Web quedaría representada mediante una red (o grafo) semántica con relaciones entre conceptos y entidades (mediante propiedades de pertenencia, de similitud, etc.), instanciados en diferentes ontologías. Dicha red facilitaría la comprensión y la exploración de los datos de la Web de una forma actualmente inabarcable. En una situación idílica en la que todos los datos estuvieran anotados y categorizados de acuerdo a unas ontologías conocidas, un sistema informático podría sustituir o asistir al usuario en tareas de búsqueda complejas, pudiendo abarcar más información en menor tiempo y con mejores resultados.

En este contexto, de entre las planteadas implementaciones de la Web Semántica, una ha sido la que más auge ha tomado y la que finalmente parece que se quedará vigente: la iniciativa conocida como Linked (Open) Data[2][6]. En vez de anotar los contenidos Web con ontologías ad-hoc no consensuadas por todos los usuarios de la Web, Linked Data plantea que se vayan estableciendo de facto un conjunto público de ontologías y bases de conocimiento que permitan la clasificación del contenido de la Web y que, de manera muy importante, se enlacen entre sí relacionando conceptos y entidades y reutilizando atributos y relaciones de todas ellas.

En la Figura se muestra un grafo con las bases de conocimiento ontológicas que forman parte de Linked Data a fecha de agosto de 2014. Los colores de los nodos de la figura reflejan las temáticas y dominios cubiertos por las bases de conocimiento (algunas se centran en dominios concretos mientras que otras son de propósito general) y los arcos indican que existen relaciones explícitas entre pares de bases de conocimiento.

De todas las bases de conocimiento que forman la "nube" de Linked Data (del inglés, Linked Data cloud), una se considera como la más importante y núcleo de la misma: DBpedia; como se puede apreciar en el centro de la figura, además de ser la de mayor tamaño, se relaciona con una gran cantidad de bases de conocimientos diversas. DBpedia es una ontología y base de conocimiento generada automáticamente a partir de información estructurada de Wikipedia[3] (esto es: cuadros de información, tablas, listados de categorías, hipervínculos, entre otros) [7].

---

[1] En Ciencias de la Computación y de la Información, una ontología es una definición formal de tipos, propiedades, y relaciones entre entidades que realmente o fundamentalmente existen para un dominio de discusión en particular, https://es.wikipedia.org/wiki/Ontología_(informá1tica)

[2] http://linkeddata.org

[3] https://www.wikipedia.org

Los motivos del éxito de DBpedia sobre otras bases de conocimiento radican en su amplia cobertura de los temas y dominios existentes, y de su constante evolución paralela a la de Wikipedia, una de las bases de conocimiento no estructurado más completa de la actualidad. Además, DBpedia, al ser una base de conocimiento estructurada –sigue el estándar RDF[4] de representación de conocimiento de la Web Semántica–, permite ejecutar consultas formales mediante lenguajes basados en SQL, como SPARQL[5]. Por ejemplo, para conocer el valor de una propiedad de una determinada entidad, habría que obtener una tripleta [`sujeto – propiedad – valor`] en la que habría que extraer el campo `valor`, y en la que `sujeto` y `propiedad` serían conocidas. Así, para conocer el año de creación (propiedad `dbp:established`) de la Universidad Autónoma de Madrid (cuya URI en DBpedia es http://dbpedia.org/resource/Autonomous_University_of_Madrid), la consulta SPARQL pertinente sería:

```
SELECT ?x WHERE {
     dbr: Autonomous_University_of_Madrid dbp:established ?x .
}
```

donde `dbr` es la abreviatura de `dbpedia.org/resource/` y `x` la variable en la que retornará el resultado de la consulta (en este caso, 1968).

En este trabajo se plantea usar DBpedia como base de conocimiento a usar por el Sistema de Búsqueda de Respuestas de dominio abierto.

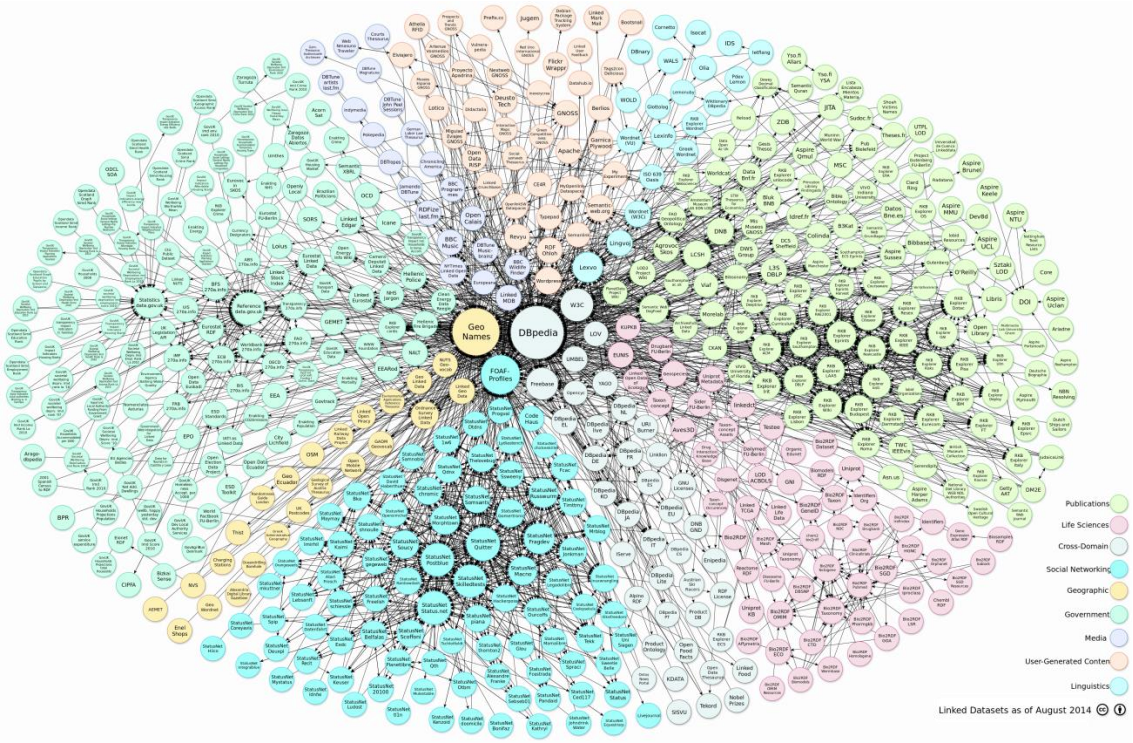

Figura 2. Ontologías y bases de conocimiento que forman Linked Data, a agosto de 2014 (Fuente: http://linkeddata.org).

[4] RDF, Resource Descriptor Framework. (https://www.w3.org/RDF)

[5] SPARQL, query lenguage for RDF (https://www.w3.org/TR/rdf-sparql-query)

1.2 Objetivos

De la lo expuesto en la sección anterior se extrae el siguiente objetivo principal de este trabajo:

El desarrollo de un Sistema de Búsqueda de Respuestas de dominio abierto planteado usando DBpedia como base de conocimiento.

La motivación/dificultad del trabajo es doble. Por una parte, Question Answering es un problema de investigación complejo abierto, no resuelto satisfactoriamente, especialmente cuando se intentan tratar preguntas no restringidas sintácticamente y sobre múltiples dominios. Por otra parte, el uso de DBpedia como base de conocimiento en un sistema de Question Answering no ha sido abordado exhaustivamente, especialmente porque hasta la fecha los sistemas no ontológicos han sido los más populares y estudiados en la literatura.

Desde el principio tiene que quedar claro que la alta dificultad y ambición del objetivo planteado, unido a la limitación de tiempo disponible, hacen que el sistema desarrollado sea un prototipo con muchas de sus componentes mejorables y ampliables, y cuya evaluación queda pendiente para un futuro.

1.3 Organización del documento

El resto del documento se estructura como sigue:

- En el apartado 2 se describen sistemas existentes de Question Answering y sus similitudes y diferencias con el desarrollado para este trabajo.
- En el apartado 3 se detallan los recursos empleados para su realización, detallando su definición y las propiedades que han resultado de especial utilidad.
- En el apartado 4 se detalla el proyecto llevado a cabo.
- En el apartado 5 se muestran ejemplos de ejecución y se especifican pruebas preliminares durante el desarrollo del sistema.
- Por último, en el apartado 6 se extraen conclusiones de la realización del trabajo y se proponen mejoras futuras.

2 Estado del arte

Pese a existir marcadas diferencias entre las distintas implementaciones de los sistemas de Question Answering, hay algunos principios básicos comunes a todos ellos. Entre ellos, destaca la necesidad de realizar un procesado de la pregunta introducida por el usuario en busca de elementos que identifiquen la respuesta buscada, la categorización de los distintos tipos de pregunta existentes –preguntas sí/no, preguntas por un lugar (*where*), preguntas por una persona (*who*), etc. – y en consecuencia, la identificación de las categorías a las que pertenecen las entidades de respuesta de las preguntas.

En esta sección se presenta una breve revisión del estado del arte en sistemas de QA, comentando cómo se abordan los aspectos anteriores por implementaciones existentes. A partir de aquí, los sistemas expuestos se diferenciarán en función de la base de conocimiento empleada. En particular, se distinguen sistemas que emplean bases de conocimiento que no pertenece a la Web Semántica, y otros que sí, en particular a la iniciativa Linked Data.

2.1 Sistemas de búsqueda de respuestas no basados en Linked Data

En los sistemas de Búsqueda de Respuesta cuya fuente de conocimiento no está basada en la Web Semántica, aparecen dos alternativas: sistemas que emplean una base de datos relacional y sistemas sobre texto libre.

Los *sistemas que emplean bases de datos relacionales* suelen estar limitados en la amplitud del dominio cubierto y en la dificultad y coste de mantener la base de datos, normalmente de gran tamaño. Por el contrario, suelen ser sistemas muy precisos que, pese a estar focalizados en un único tema, suelen cubrir bastante bien todas las posibles preguntas al respecto. Es el caso del sistema Baseball [2], uno de los primeros sistemas de Question Answering desarrollados, que consiguió una gran precisión y amplitud en cuanto a las preguntas soportadas, siempre referentes al deporte que lleva por nombre.

Los *sistemas basados en texto libre* utilizan como fuente documentos planos y páginas de internet no necesariamente redactadas y preparadas para dicho fin. Cualquier documento existente es, por tanto, válido para ser incluido en la base de conocimiento, lo que aporta una enorme flexibilidad y capacidad de aprendizaje al sistema.

Según un reciente artículo de Hirschman [1], un sistema de búsqueda de respuestas de este tipo consta de las siguientes etapas:

1. **Análisis de la pregunta**: división en partes de interés o bloques semánticos, contextualización e hipótesis. El objetivo de esta fase es la obtención de distintas representaciones de la pregunta que puedan ser analizadas de manera independiente más adelante (a mayor número de interpretaciones, más posibilidades de encontrar la respuesta). Estas interpretaciones tendrán asociado un peso en función de la "confianza" que tiene el sistema sobre las hipótesis del significado de la pregunta que plantean.
2. **Pre-procesado de la colección de documentos**: análisis de los documentos que conforman la base de conocimiento con el objetivo de calcular la probabilidad de encontrar la respuesta deseada en cada uno de ellos. El objetivo de esta fase es obtener una representación lógica de los documentos analizados, etiquetando las

entidades de interés e indexando la información que se considera relevante. Se trata, eso sí, de un análisis muy superficial.

3. **Selección de documentos en base a las coincidencias existentes entre las fases anteriores**: se comparan los documentos anteriormente procesados con las distintas hipótesis obtenidas en el análisis de la pregunta inicial y, en caso de existir información coincidente, son marcados como documentos candidatos.
4. **Análisis de los documentos candidatos seleccionados**: realiza un análisis más profundo de los documentos, haciendo especial énfasis en los párrafos o segmentos de los textos en los que se localizan las coincidencias con la pregunta del usuario. Se puntúa la precisión de la coincidencia y se otorga un peso a las respuestas ofrecidas por cada candidato.
5. **Extracción de soluciones a la pregunta y generación de la respuesta al usuario**: se estructura la solución a la pregunta en una oración.

Un ejemplo de sistema basado en texto libre es Watson [8], de IBM, que analiza la sentencia en busca de "pistas" que convierte en distintas búsquedas entre sus documentos. Dicha búsqueda se divide en 50 patrones paralelos, seguidos de una fase final de selección de la respuesta más probable.

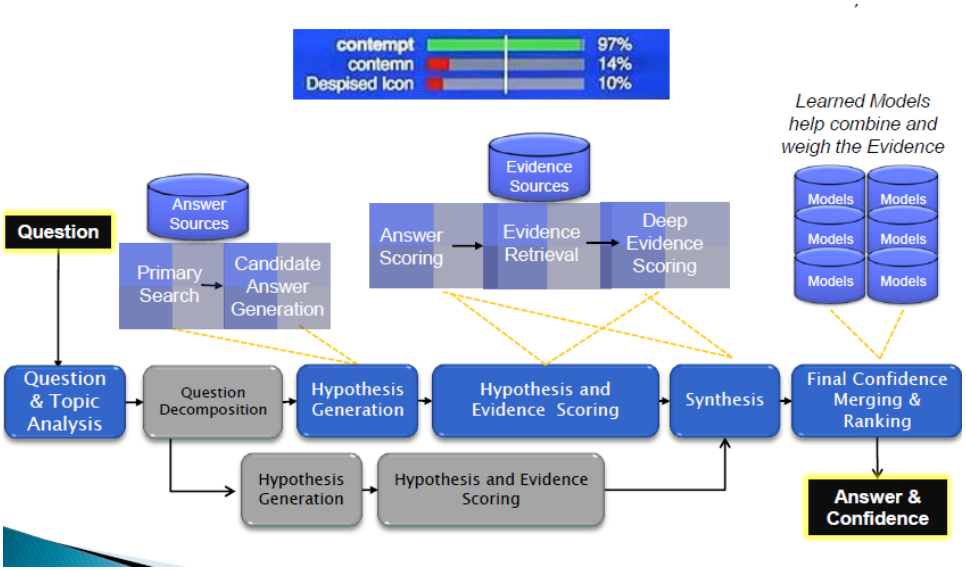

Figura 2. Estructura de Watson [12].

2.2 Sistemas de búsqueda de respuestas basados en Linked Data

Antes de comenzar a analizar distintos sistemas existentes basados en Linked Data, es importante resaltar que en la Web Semántica cada recurso aparece identificado por un identificador único, URI, a partir del cual se puede acceder a sus propiedades y relaciones, lanzando consultas sobre la base de conocimiento correspondiente, usando lenguajes de consulta formales, como SPARQL. Por ello, una de las fases vitales de estos sistemas será la relación entre el texto plano introducido por el usuario y patrones de consultas formales.

PowerAqua [9], sistema pionero en el uso de ontologías para abordar el problema de QA, se basa en la búsqueda de coincidencias (*matching*): a partir de los distintos elementos extraídos en el análisis de la pregunta, se buscan los recursos "sinónimos" en la base de conocimiento (existe una relación entre objeto en lenguaje natural y URI del objeto en la

base de conocimiento). Para intentar maximizar el éxito de dicha relación, se hace uso de WordNet para la extracción de sinónimos y de similitudes de cadenas de cadenas de caracteres para buscar los elementos que más se asemejan al recurso que estamos buscando.

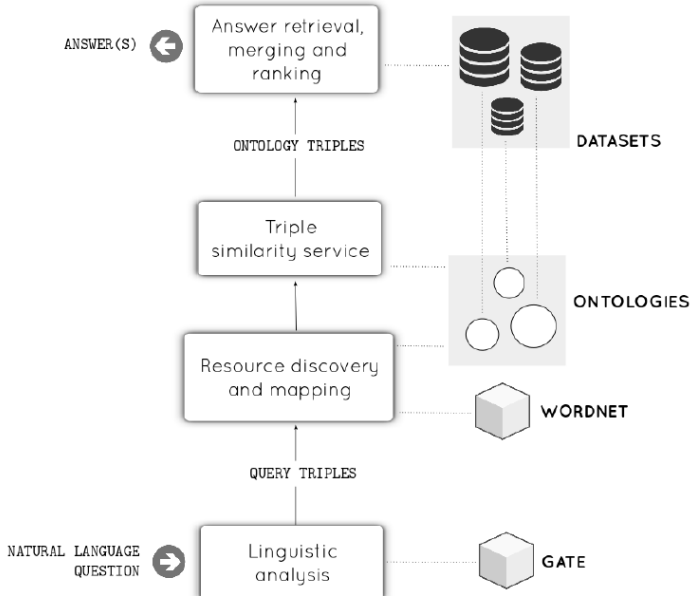

Figura 3. Estructura de PowerAqua [12].

Orakel [10] por su parte, hace uso de ontologías para analizar la pregunta introducida por el usuario. Distingue entre palabras independientes (artículos, pronombres, preposiciones y cualquier otra palabra cuyo significado no sea un recurso) y palabras específicas, esto es, que se refieren a recursos conocidos por Linked Data. Una vez conocidos los recursos, se intenta encontrar la respuesta buscada en función de la disposición de las demás palabras.

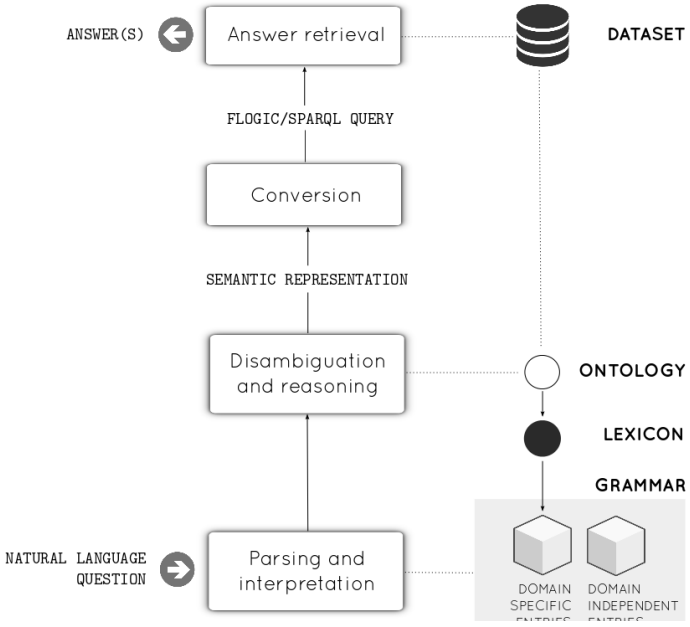

Figura 4. Estructura de Orakel [12].

TBSL [11] es un sistema basado en la generación de estructuras de consultas que coincidan directamente con estructuras de preguntas conocidas. De esta forma, si se introduce una pregunta incluida en el sistema, la consulta asociada determinará qué es cada elemento de la pregunta y lanzará la consulta directamente.

Como se puede apreciar, las diferencias fundamentales entre los sistemas descritos residen en la forma en que se lleva a cabo el análisis de la pregunta recibida y en el momento (fase) en que se utiliza la base de conocimiento. En el sistema desarrollado para este trabajo, hay muchas similitudes con PowerAqua, ya que el proceso de utilización de sinónimos para buscar coincidencias en DBpedia resulta muy útil para maximizar coincidencias sin que el sistema dependa del lenguaje empleado por el usuario. Sin embargo, a diferencia de los anteriores, se ha decidido realizar un análisis sintáctico previo, con el objetivo de identificar los distintos recursos y focalizar la búsqueda en las categorías pertinentes, en función de si hacen referencia a nombres propios (ya sea una persona, un lugar o una entidad), a propiedades de un determinado recurso (fecha de nacimiento, ocupación, etc.) o a tipos a los que pertenece el sujeto.

Este análisis sintáctico, además, es usado para la identificación de tipos de preguntas, basándose en la presencia y orden de los distintos sintagmas que componen la oración, siendo capaces de identificar -pudiendo generar consultas sobre patrones de preguntas no determinados de manera concreta previamente- si éstas coinciden con la estructura sintáctica de algún tipo conocido. De esta forma, el sistema propuesto es capaz de reconocer e intentar dar respuesta a tipos de preguntas de manera dinámica y más flexible que TBSL u Orakel, que requieren una coincidencia exacta, bien sea de los recursos que aparecen en la oración o de la propia estructura de la sentencia.

3 Tecnologías y recursos empleados

En esta apartado se detallan las tecnologías, herramientas y recursos utilizados para el desarrollo del sistema presentado dividiéndose en dos grupos:

- Recursos sobre los que se apoya la fuente de conocimiento del sistema, en este caso DBpedia, con las facilidades y características que ofrecen las tecnologías de Web Semántica y la iniciativa Linked Data.
- Herramientas y recursos utilizados para el análisis del texto de entrada al programa, entre las que se incluye el análisis sintáctico de la pregunta (Librería PLN de la Universidad de Stanford), la búsqueda de sinónimos (diccionario WordNet de la Universidad de Princeton) para maximizar las probabilidades de encontrar la información solicitada, y la identificación de patrones sintácticos en dichas preguntas, que permitan la traducción de dichas preguntas en consultas formales válidas para la base de conocimiento.

3.1 Web Semántica

El enorme impacto de la World Wide Web en la sociedad ha hecho que su crecimiento haya sido exponencial en las últimas décadas. Tanto es así, que supera en cantidad de información a las bibliotecas más importantes del mundo [13], lo que hace que la navegación manual por ella se vuelva costosa e ineficiente.

El principio de la Web Semántica pretende facilitar el acceso y la navegación a máquinas autónomas, con inteligencia artificial, que puedan sustituir a los humanos en la ardua tarea que supone la búsqueda, clasificación y verificación de la información en la Web. Para ello, se propone la introducción de descripciones precisas de los recursos que conforman la web, dando lugar a una web más cohesionada, en la que sea más sencillo integrar información y servicios, optimizando los recursos disponibles en ella [4].

La transformación propuesta por este principio es costosa: supone modificar la base de la Web, sustituyendo el enfoque actual (en el que la información está pensada para un público humano y los lenguajes de programación usados sólo contienen información sobre la estética de la Web y no sobre los conceptos tratados) a un enfoque basado en la conexión de la información a través de toda la Web, manteniendo la cohesión de los datos y meta información de los recursos. La figura 6 representa gráficamente esta diferencia: las referencias a otros recursos en la web actual se hace mediante la etiqueta genérica `href` mientras en la Web Semántica dichas referencias mostrarían explícitamente el tipo de relación existente entre ambos objetos.

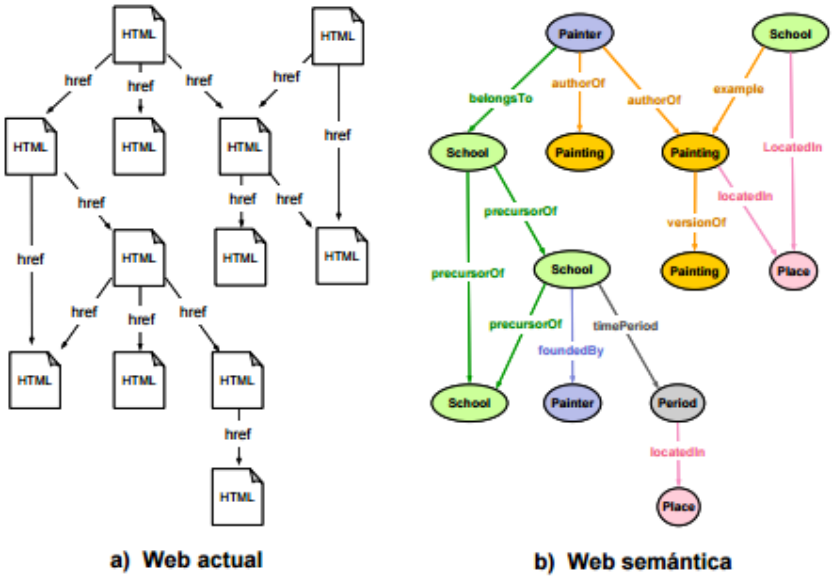

Figura 5. Comparativa de la Web actual y la Web semántica [4].

El lenguaje usado para la representación de conocimiento en la Web Semántica es RDF (Resource Description Framework), que permite la definición de ontologías y datos (instancias) a base de tripletas [`sujeto, propiedad, objeto`], donde cada elemento de la relación es un recurso de la Web Semántica con una URI que lo identifica y con atributos y relaciones a otros recursos asociadas. Esta representación suele hacerse mediante XML, siendo cada etiqueta una propiedad, como puede observarse a continuación:

```
<dbpedia-owl:country rdf:resource="http://dbpedia.org/resource/Spain" />
<dbpedia-owl:demonym xml:lang="es">madrileño, -ña o</dbpedia-owl:demonym>
<dbpedia-owl:demonym xml:lang="es">matritense</dbpedia-owl:demonym>
<dbpedia-owl:flag>Bandera de Madrid.svg</dbpedia-owl:flag>
<dbpedia-owl:municipalityCode>079</dbpedia-owl:municipalityCode>
<dbpedia-owl:postalCode>28001-28080</dbpedia-owl:postalCode>
<dbpedia-owl:province rdf:resource="http://dbpedia.org/resource/Community_of_Madrid" />
<dbpedia-owl:saint rdf:resource="http://dbpedia.org/resource/Isidore_the_Laborer" />
<dbpedia-owl:year rdf:datatype="http://www.w3.org/2001/XMLSchema#gYear">2014+01:00</dbpedia-owl:year>
<foaf:name xml:lang="es">Madrid</foaf:name>
<ns9:areaTotal rdf:datatype="http://dbpedia.org/datatype/squareKilometre">605.0</ns9:areaTotal>
```

Figura 7. Estructura RDF de Madrid en DBpedia.

La consulta sobre bases de conocimiento en RDF se realiza en general a través de lenguajes formales similares en SQL, como SPARQL. A través de estos lenguajes de consulta se puede recuperar información de una Web "estructurada" que permita abordar las limitaciones existentes en la recuperación de información Web tradicional, tales como la comprensión de la semántica subyacente a una consulta y la desambiguación de conceptos involucrados. El siguiente ejemplo muestra una consulta del valor de la propiedad `dbo:country` (país) de Madrid (cuyo RDF se muestra en la figura 7):

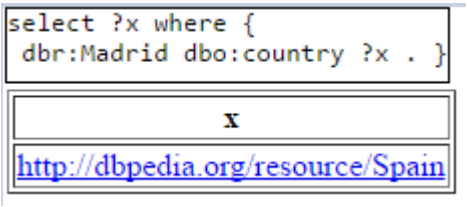

Figura 8. Ejemplo de consulta y respuesta en SPARQL

3.2 Linked Data y DBpedia

Dentro del contexto de la Web Semántica, Linked Data pretende la creación de una red de ontologías y bases de conocimiento estructuradas que enlacen y compartan información públicamente, garantizando el acceso de cualquier usuario (humano o máquina) a un conjunto de datos cohesionado.

Se basa en 4 principios[6]: 1) usar identificadores únicos, URIs, para los recursos (conceptos, objetos, personas, organizaciones, etc.); 2) hacer que dichas URIs sean accesibles mediante HTTP, para permitir a los usuarios acceder a los recursos vía Web; 3) cuando alguien acceda a una URI, proporcionar toda la información disponible usando estándares establecidos, como RDF; 4) hacer que las propiedades de recursos y enlaces a otros deben ser también URIs, con el fin de permitir la navegación entre ellos.

Cada una de las ontologías y bases de conocimiento que forman Linked Data está conectada con las demás de manera que comparten información. Éstas pueden estar focalizadas en un área de conocimiento o dominio concreto o ser de ámbito general. Los enlaces entre ellas permiten ampliar los recursos accesibles desde cada una.

Dentro de la red de Linked Data, DBpedia ocupa un papel central. Se trata de una base de conocimiento de ámbito abierto que pretende la extracción de información de la enciclopedia en línea Wikipedia, además de la integración con otras fuentes de Linked Data. En la figura 9 se pueden observar algunas conexiones de DBpedia con otras fuentes de Linked Data con información sobre diversos dominios. Por ejemplo, FOAF (Friend Of A Friend, en sus siglas en inglés) está diseñada para la descripción de personas y sus relaciones con otras, Geo-names contiene información geográfica, y RDF Book Mashup está centrada en publicaciones literarias.

[6] https://www.w3.org/DesignIssues/LinkedData.html

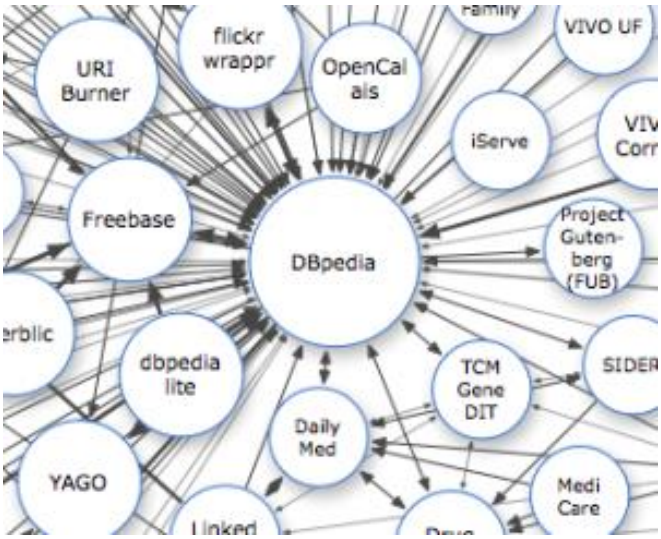

Figura 9. Algunos de las bases de conocimiento enlazadas con DBpedia. (Fuente: https://es.wikipedia.org/wiki/DBpedia)

Según su página Web[7], a fecha de mayo de 2016, la versión inglesa de DBpedia cuenta con 4.58 millones de elementos descritos, de los cuales 4.22 millones están clasificados en una red ontológica, de los cuales hay 1.45 millones de personas clasificadas, 735.000 lugares geográficos, 87.000 películas, 6.000 enfermedades, etcétera. Toda esta información está disponible en 125 idiomas (con ligeras variaciones, como ocurre con las páginas de mismo recurso en DBpedia; la más completa es la inglesa).

A continuación, en las figuras 10 y 11, se muestra parte del recurso RDF en inglés de la Universidad Autónoma de Madrid, cuya URI es http://dbpedia.org/page/Autonomous_University_of_Madrid. En la vista Web del recurso, se listan las propiedades conocidas del objeto con su valor. Algunas de las más relevantes de las que aparecen en las imágenes son `dbp:abstract`, que incluye un pequeño resumen descriptivo de la entidad; `dbo:affiliation`, que muestra categorías a las que la universidad pertenece (en este caso, redirige a los recursos propios de la Asociación Europea de Universidades y a la red institucional de universidades de capitales europeas); `dbo:city`, que indica la ciudad en la que se encuentra, etc. Como puede verse, algunas de las propiedades dependerán de la categoría de la entidad: al ser un objeto de lugar, tiene propiedades como ubicación o país, que entidades de tipo persona no tendrán. A cambio, ésta tendrá otras propias de su tipo como el lugar de nacimiento o de muerte.

[7] http://wiki.DBpedia.org/About

Figura 10. Valor de la propiedad ‘abstract’ (resumen) del recurso DBpedia asociado a la Universidad Autónoma de Madrid.

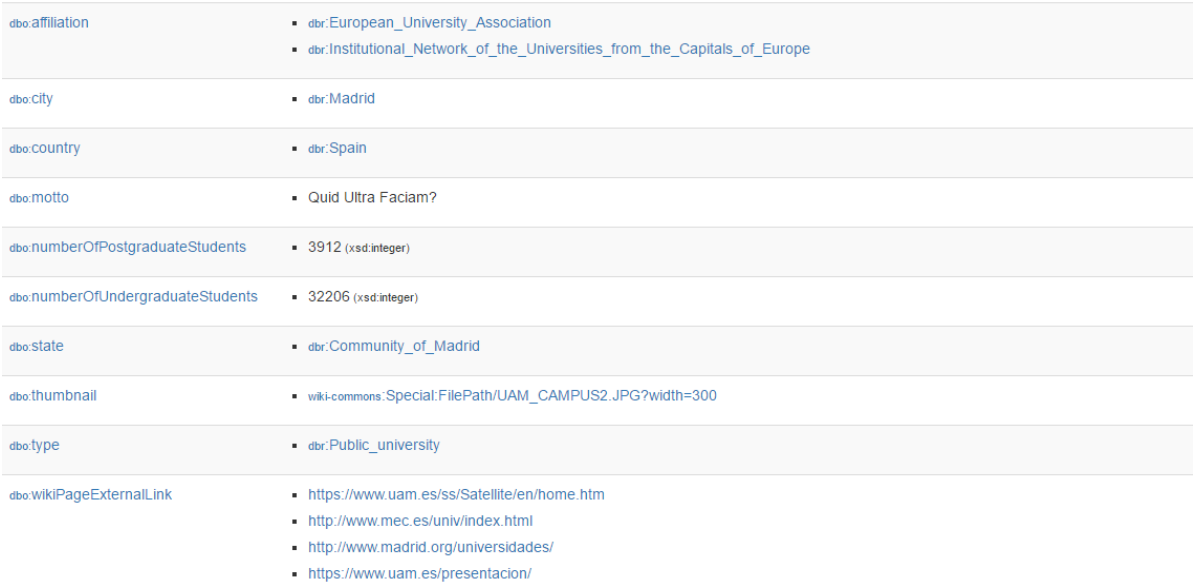

Figura 11. Valores de varias propiedades del recurso DBpedia asociado a la Universidad Autónoma de Madrid.

Si se quisiera conocer una propiedad concreta de la entidad (recurso) ‘Universidad Autónoma de Madrid’, deberíamos formar una tripleta en lenguaje SPARQL. Para ello, indicaremos [ENTIDAD]-[PROPIEDAD DESEADA]-[X], donde X será el retorno de la consulta. Por ejemplo, si se desease conocer la ubicación de esta entidad (representada por la propiedad ‘dbo:state’) la consulta sería:

```
SELECT ?x WHERE {
     dbr:Universidad_Autonoma_Madrid dbo:state ?x .
}
```

La solución de dicha consulta será otro recurso accesible, que a su vez tendrá una serie de propiedades definidas.

x
http://dbpedia.org/resource/Community_of_Madrid

Figura 12. Respuesta de la consulta SPARQL.

3.3 Stanford CoreNLP

Una de las partes fundamentales del Question Answering consiste en el análisis (sintáctico) de la pregunta del usuario para la identificación de los distintos recursos que están mencionados, y a ser usados en las búsquedas de información posteriores.

En el sistema desarrollado el análisis sintáctico de las preguntas de usuario se realiza mediante la herramienta CoreNLP, de Procesado de Lenguaje Natural, de la Universidad de Stanford [http://nlp.stanford.edu/software/], que permite distintas opciones de configuración para el análisis de oraciones. A continuación se detallan las más relevantes:

1. ***Tokenization***: división del texto de entrada en palabras.
2. ***Lemmatization***: identificación de la raíz (morfema) de las palabras. Si es un verbo conjugado, su *lemma* será el verbo en infinitivo.
3. ***Part-Of-Speech (POS) recognition***: identificación de la categoría gramatical o morfosintáctica a la que pertenece una palabra: nombre, verbo, adjetivo, etc.
4. ***Named Entity Recognition (NER)***: determinación de si un palabra pertenece a una entidad con nombre propio: persona, organización, lugar, etc.
5. ***Parsing***: lleva a cabo el análisis sintáctico de la oración, dando como resultado un árbol gramatical en el que se identifican las dependencias entre las distintas palabras y grupos sintácticos.

A continuación se muestran algunos ejemplos de un programa sencillo que analiza (*parsea*) una oración y extrae la información relativa a cada una de las opciones mentadas anteriormente. Las abreviaturas de POS[8] especifican el tipo de palabra que es (artículo, determinante, nombre…); las anotaciones, hacen referencia a si un elemento (*token*) analizado es una persona, una ubicación o una empresa (en este caso, ninguna de las palabras lo es, por lo que aparecen vacías); el sentimiento indica la inclinación de opinión o cualidad, positiva o negativa, de las palabras (en este caso, todas ellas son etiquetadas como neutras). Finalmente, se muestra el sentimiento identificado en la oración y un análisis sintáctico descompuesto en una estructura de árbol.

[8] http://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

```
Sentence: I will make an offer you can't refuse
Tokenize        Lemma   POS     Anotation       Sentiment
I               I       PRP     O               Neutral
will            will    MD      O               Neutral
make            make    VB      O               Neutral
an              a       DT      O               Neutral
offer           offer   NN      O               Neutral
you             you     PRP     O               Neutral
ca              can     MD      O               Neutral
n't             not     RB      O               Neutral
refuse          refuse  VB      O               Neutral
Sentence sentiment: Positive
Sentence tree:
(ROOT
  (S
    (NP (PRP I))
    (VP (MD will)
      (VP (VB make)
        (NP
          (NP (DT an) (NN offer))
          (SBAR
            (S
              (NP (PRP you))
              (VP (MD ca) (RB n't)
                (VP (VB refuse)))))))))
```

Figura 13. Valor de las etiquetas descritas en una frase de ejemplo.

En el sistema desarrollado, las anotaciones relevantes son las del *parser* y el *tokenizer* ya que a partir del análisis sintáctico llevado a cabo por la herramienta de Stanford, se procesará de nuevo la pregunta de acuerdo a los intereses concretos de identificación de recursos. A pesar de que el proyecto incluye otros idiomas, la versión más avanzada y precisa es la inglesa, estando la española todavía en fases tempranas de desarrollo.

3.4 WordNet

WordNet es una base de datos de léxico que proporciona definiciones de términos, búsqueda de sinónimos y las relaciones semánticas con otros conceptos [https://wordnet.princeton.edu]. Su propósito principal es su uso e integración con sistemas de inteligencia artificial basadas en análisis de textos, además de su uso como diccionario, fácilmente accesible por API que proporciona.

La figura 14 muestra parte de la información proporcionada por la versión web de WordNet sobre el término “car”:

Noun

- S: (n) **car**, auto, automobile, machine, motorcar (a motor vehicle with four wheels; usually propelled by an internal combustion engine) *"he needs a car to get to work"*
- S: (n) **car**, railcar, railway car, railroad car (a wheeled vehicle adapted to the rails of railroad) *"three cars had jumped the rails"*
- S: (n) **car**, gondola (the compartment that is suspended from an airship and that carries personnel and the cargo and the power plant)
- S: (n) **car**, elevator car (where passengers ride up and down) *"the car was on the top floor"*
- S: (n) cable car, **car** (a conveyance for passengers or freight on a cable railway) *"they took a cable car to the top of the mountain"*

Figura 14. Información de WordNet sobre el término "car".

Cada una de las entradas corresponde con una acepción diferente para coche (*synsets*). Cada una de ellas, contiene además sinónimos para el significado indicado, ejemplos de uso del término en dicho contexto, "hipónimos" e "hiperónimos".

La API funciona de manera similar, pudiendo acceder a esta información con distintos métodos.

4 Sistema desarrollado

4.1 Visión General

Como cualquier sistema de Question Answering de los vistos en el apartado de Estado del Arte, el sistema desarrollado en este trabajo puede subdividirse en tres partes: análisis de la pregunta introducida por el usuario en lenguaje natural, correlación de la información extraída en dicha fase con recursos conocidos por la base de conocimiento y consulta sobre ésta última en busca de la respuesta.

A continuación se muestra un esquema de las fases que se siguen en el sistema para dar respuesta a una pregunta:

QUESTION	SYNTACTIC ANALYSIS	MAPPING	QUERY	OUTPUT

Which companies were founded by Steve Jobs?	- TYPE TOKENS: [WDT/Which, NNS/companies]
	- TYPE PATTERN: [WHICH, NOUN]
	- VERB: were founded
	- VERB TOKENS: [VBD/were, VBN/founded]
	- SUBJECT:
	- SUBJECT TOKENS: []
	- SUBJECT COMPLEMENTS: by Steve Jobs
	- SUBJECT COMPLEMENTS TOKENS: [IN/by, NNP/Steve, NNP/Jobs]
		[WHAT TYPE PROPERTY OBJECT]
		X <rdf:type> Company
		X <founded> Steve Jobs
		
		Companies = dbo:Company
		
		Were founded = dbo:foundedBy
		
		http…/Steve_Jobs
			SELECT ?x WHERE {
			?x dbo:foundedBy <http://dbpedia.org/resource/Steve_Jobs> .
			}
				Apple Inc.
				NeXT

Figura 15. Fases de procesamiento de una pregunta.

En la fase de análisis de lenguaje natural (Syntactic Analysis), se ha optado por llevar a cabo un análisis sintáctico de la oración en busca de patrones de estructuración de preguntas. Se han desarrollado reglas de decisión para cada tipo de pregunta en función de cómo llega ordenada sintácticamente la información, identificando además el tipo de recurso que aporta cada sintagma. Para ello, se ha usado la librería CoreNLP de la Universidad de Stanford y unas clases propias que, a partir del árbol sintáctico, extraen y ordenan la información para procesarla de acuerdo a los diagramas de decisión citados anteriormente. La salida de esta fase es múltiple: el tipo de pregunta y las entidades (nombres propios y comunes) y propiedades (procedentes de verbos, adjetivos) involucrados.

Una vez llevada a cabo la identificación de recursos en la pregunta introducida, se "mapean" los distintos objetos con los recursos (entidades y propiedades) de DBpedia que correspondan; para ello, se hace uso de una base de datos implementada de forma que relaciona los posibles objetos en lenguaje natural con su equivalente en DBpedia, en función del tipo de recurso (categoría, entidad, propiedad, etc.).

Por último, con el tipo de pregunta y recursos identificados se construye una consulta SPARQL y se lanza sobre DBpedia, usando los identificadores obtenidos tras el mapeo. La consulta lanzada dependerá de la estructura sintáctica identificada en la primera fase, variando en función del tipo de pregunta y el orden en el que aparece la información en ella.

Adicionalmente, se ha desarrollado un módulo que identifica preguntas compuestas (en las que aparecen conectores lógicos como AND u OR) y lo divide en preguntas simples que son analizadas por separado y cuyas respuestas son tratadas con el conector lógico que se indique antes de mostrar la información al usuario.

Finalmente, se ha desarrollado una interfaz para la interacción con el usuario por terminal, en el que la respuesta obtenida de la consulta es mostrada al usuario en lenguaje natural. Asimismo, existe un módulo de corrección de errores que pretende evitar que se pare la ejecución del sistema por problemas tipográficos y similares, que pueden ser solucionados por el usuario en otra iteración. Por ejemplo, introducir "Ral Madrid" en vez de "Real Madrid" causaría que no se encontraran coincidencias en el mapeo de la entidad y ocasionaría un error al que se intentará dar solución volviendo a preguntar al usuario a qué se refiere con ello. La comunicación entre interfaz-gestor de errores-sistema, así como el funcionamiento detallado de cada módulo, se explica en los apartados siguientes.

4.1.1 Identificación del patrón de pregunta

El funcionamiento del sistema al completo depende del buen análisis de la entrada recibida ya que de esta fase depende tanto la identificación de los recursos como la selección de la plantilla de consulta que se empleará para localizar la respuesta. Esta tarea se puede subdividir a su vez en dos: un análisis previo llevado a cabo mediante el sistema NLP de la Universidad de Stanford y un tratado del árbol sintáctico resultante, para identificar los recursos disponibles y el tipo de pregunta a resolver.

Para ello, se ha desarrollado una clase cuyo propósito es la clasificación de la información obtenida en el análisis sintáctico de la oración en estructuras que identifiquen el tipo de recurso al que hacen referencia, distinguiéndose *Verbo, Sujeto, Complementos del sujeto, Objeto, Complementos del objeto* y *Tipo*. En *Verbo* se guarda la estructura verbal principal de la oración; *Sujeto* almacena el sujeto de la oración y sus *Complementos* irán a la estructura del mismo nombre. En *Tipo* se guardará toda la información que aparezca entre la palabra clave de la pregunta (*what, where, when*…) y el verbo principal de la oración**.** Las estructuras *Objeto* y *Complemento de Objeto* se han utilizado como sinónimo de las estructuras de sujeto, analizando estructuras sintácticas complementarias.

En función de cómo queda repartida la información en las citadas estructuras, se han elaborado diagramas de decisión que gestionan el proceso en cada caso. Se ha trabajado sobre un conjunto muy variado de preguntas -en lo referente a la estructura sintáctica-, con el fin de identificar el mayor número de patrones posible y así dar respuesta a un mayor número de cuestiones.

QUESTION	SYNTACTIC OUTPUT
Which football players have played in Real Madrid and Barcelona?	- TYPE TOKENS: [WDT/Which, NN/football, NNS/players] - TYPE PATTERN: [WHICH, NOUN, NOUN] - VERB: have played - VERB TOKENS: [VBP/have, VBN/played] - OBJECT: in Real Madrid and Barcelona - OBJECT TOKENS: [IN/in, JJ/Real, NNP/Madrid, CC/and, NNP/Barcelona]
Which 1997 films were starred by Kate Winslet?	- TYPE TOKENS: [WDT/Which, CD/1997, NNS/films] - TYPE PATTERN: [WHICH, DETERMINER, NOUN] - VERB: were starred - VERB TOKENS: [VBD/were, VBN/starred] - OBJECT: by Kate Winslet - OBJECT TOKENS: [IN/by, NNP/Kate, NNP/Winslet]
What novelists or writers founded Naughty Dog?	- TYPE TOKENS: [WDT/What, NNS/novelists, CC/or, NNS/writers] - TYPE PATTERN: [WHAT, NOUN, CONNECT, NOUN] - VERB: founded - VERB TOKENS: [VBN/founded] - OBJECT: Naughty Dog - OBJECT TOKENS: [NNP/Naughty, NNP/Dog]
Which American presidents were born in Massachusetts or in Hawaii?	- TYPE TOKENS: [WDT/Which, JJ/american, NNS/presidents] - TYPE PATTERN: [WHICH, ADJECTIVE, NOUN] - VERB: were born - VERB TOKENS: [VBD/were, VBN/born] - OBJECT: in Massachussetts or in Hawaii - OBJECT TOKENS: [IN/in, NNP/Massachussetts, CC/or, IN/in, NNP/Hawaii]
What city is Hollywood located in?	- TYPE TOKENS: [WDT/What, NN/city] - TYPE PATTERN: [WHAT, NOUN] - VERB: is located - VERB TOKENS: [VBZ/is, VBN/located] - SUBJECT: Hollywood - SUBJECT TOKENS: [NNP/Hollywood]
Which football players has Sara Carbonero been married with?	- TYPE TOKENS: [WDT/Which, NN/football, NNS/players] - TYPE PATTERN: [WHICH, NOUN, NOUN] - VERB: has been married - VERB TOKENS: [VBZ/has, VBN/been, VBN/married] - SUBJECT: Sara Carbonero - SUBJECT TOKENS: [NNP/Sara, NNP/Carbonero] - OBJECT: with - OBJECT TOKENS: [IN/with]
Which films of Leonardo DiCaprio have won an Oscar?	- TYPE TOKENS: [WDT/Which, NNS/films, IN/of, NNP/Leonardo, NNP/DiCaprio] - TYPE PATTERN: [WHICH, NOUN, PREPOSITIONAL PHRASE] - VERB: have won - VERB TOKENS: [VBP/have, VBN/won] - OBJECT: an Oscar - OBJECT TOKENS: [DT/an, NNP/Oscar]

Figura 16. Valor de las estructuras semánticas en distintos ejemplos.

Sin embargo, la salida de alguna de estas preguntas ha hecho imposible su correcta solución, ya que o bien parte de la información era omitida en el proceso de organización en las estructuras semánticas, o bien no se correspondía con ningún patrón conocido:

QUESTION	SYNTACTIC OUTPUT
Which American presidents are still alive?	- TYPE TOKENS: [WDT/Which, JJ/american, NNS/presidents] - TYPE PATTERN: [WHICH, ADJECTIVE, NOUN] - VERB: are - VERB TOKENS: [VBP/are]

Figura 17. Pregunta con patrón conflictivo.

En el caso anterior, por ejemplo, la condición "still alive" es obviada en el procesado, quedando una clasificación sintáctica incompleta. Además, para reducir el enorme número de patrones existentes, los tipos de pregunta estudiados parten de la premisa de que las estructuras *Sujeto* y *Objeto* son equivalentes.

Los tipos de preguntas procesadas por el sistema son WHAT, WHICH, WHERE, WHO, WHEN y preguntas de tipo YES/NO. En todos los casos, se han hecho árboles/reglas de decisión independientes, teniendo en cuenta si el verbo es "to be" o cualquier otro (es decir, un árbol para WHAT IS/ARE y otro distinto para WHAT seguido de otro verbo).

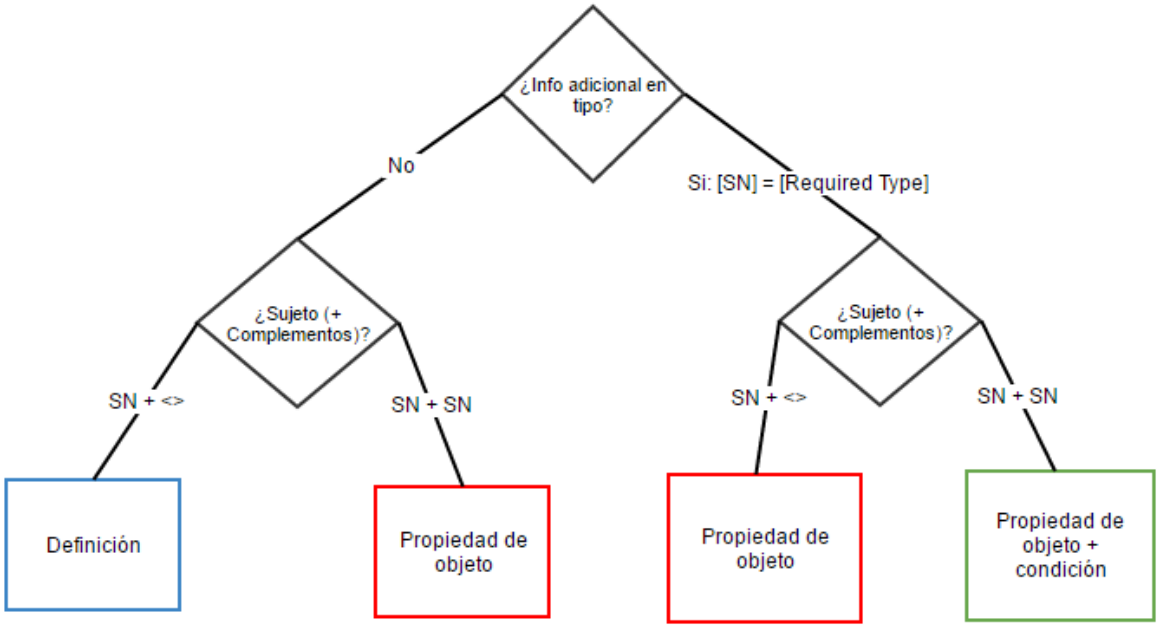

Figura 18. Diagrama de decisión del patrón WHAT-IS.

En la figura 18 se presenta el árbol de decisión del tipo de pregunta WHAT-IS. Las estructuras sintácticas empleadas aparecen identificadas como **verbo**, **tipo**, **sujeto** (equivalente a objeto) y **complementos**. Como puede apreciarse, en función de la información almacenada en cada una de las distintas estructuras, el sistema elegirá la consulta a llevar a cabo. Aunque no aparece en la imagen, dicha ramificación lleva implícita la asignación de cada estructura a un tipo de recurso.

En la rama más a la izquierda, se observa una consulta que no tiene información en el *tipo*, cuyo *sujeto* está formado por un nombre y que no tiene complementos adicionales. Dicha estructura es, por tanto, del tipo "What is [object]?", que se identifica como la definición del objeto que aparece en el *sujeto*.

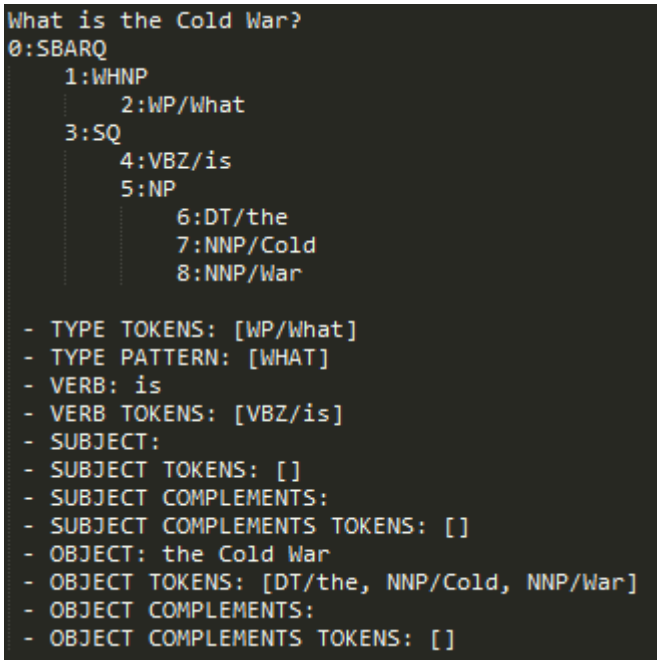

Figura 19. Procesado de pregunta de tipo *Definición*.

En dicho caso, la consulta a llevar a cabo será una definición, en la que se preguntará por la propiedad **dbo:abstract** del objeto ya que, como hemos visto antes, esa propiedad contiene una pequeña descripción del recurso. Por último, se determina que la información contenida en **object** será el sujeto de la oración (y por tanto, el recurso del que hay que obtener la entidad sobre la que realizar la consulta).

Cada rama llevará asociada una consulta distinta; si en el ejemplo se muestra el caso de definición, otras ramas llevan a "Propiedad de objeto", donde se preguntará por la propiedad deseada del objeto indicado, identificando qué estructura contiene la propiedad y cuál el sujeto.

En la salida de esta fase, se conocerá la consulta a realizar y los distintos recursos disponibles para ello. Estos recursos están en lenguaje natural, por lo que será necesario que módulos posteriores conviertan esta información en recursos válidos de DBpedia.

PREGUNTA	PLN	Identificación de recursos
Which is the capital of Spain?	WHICH-IS capital spain	WHICH-IS capital = property spain = object
Who is the president of Real Madrid?	WHO-IS presidente real madrid	WHO-IS president = property real madrid = object
What was the Cold War?	WHAT-IS cold war	WHAT-IS cold war = object

Figura 20. Salida del módulo, con la información identificada en función del tipo de recurso

Otra característica que se ha abordado es lo que se ha definido como “consultas ocultas”: se trata de información integrada en la pregunta pero no accesible de manera directa por la base de conocimiento, ya que no se trata de un recurso como tal sino que requiere un procesamiento adicional para su identificación. Un ejemplo de este hecho sería la pregunta: “What is the name of the King of Spain?” cuya estructura sintáctica y el árbol de decisión mostrado anteriormente llevarían a la identificación de una consulta del tipo “propiedad de objeto”, como se muestra a continuación:

```
What's the name of the king of Spain?
0:SBARQ
    1:WHNP
        2:WP/What
    3:SQ
        4:VBZ/'s
        5:NP
            7:DT/the
            8:NN/name
            9:PP
                10:IN/of
                13:DT/the
                14:NN/king
                16:IN/of
                18:NNP/Spain

- TYPE TOKENS: [WP/What]
- TYPE PATTERN: [WHAT]
- VERB: 's
- VERB TOKENS: [VBZ/'s]
- SUBJECT:
- SUBJECT TOKENS: []
- SUBJECT COMPLEMENTS:
- SUBJECT COMPLEMENTS TOKENS: []
- OBJECT: the name
- OBJECT TOKENS: [DT/the, NN/name]
- OBJECT COMPLEMENTS: of the king of Spain
- OBJECT COMPLEMENTS TOKENS: [IN/of, DT/the, NN/king, IN/of, NNP/Spain]
```

Figura 21. Ejemplo de consulta con Hidden Query.

Dicha identificación, llevaría al mapeado de “the name” como propiedad deseada y “the King of Spain” como sujeto. Sin embargo, “King of Spain” no hace referencia a un recurso, sino a una nueva consulta: la propiedad “king” del objeto “Spain”, cuyo resultado será el sujeto real de la pregunta, en este caso, la URI de Felipe VI.

```
Question: What's the name of the king of Spain?

[WHAT IS PROPERTY NOUN]

--Property: name  of: (king  -> Spain )
```

Figura 22. Resolución de la consulta anterior, marcando la consulta oculta.

Las consultas ocultas suelen ser de carácter muy simple y son resueltas de manera interna durante la fase de mapeado, tratadas siempre como consultas del tipo “Propiedad del objeto”, ya que en los ejemplos estudiados siempre aparece con esta estructura.

4.1.2 Procesado de preguntas compuestas

Se define como pregunta compuesta aquella en la que se solicita la relación, ya sea de unión o intersección, de determinada información de varios objetos. Esta relación viene definida por conjunciones como “and” u “or” y su procesado requiere un trato especial.

Su identificación y procesado se realiza en un módulo a parte que, al igual que el módulo de identificación de patrones, parte de las estructuras obtenidas tras el análisis sintáctico del módulo NLP de Stanford.

QUESTION	SYNTACTIC OUTPUT
What films have Brad Pitt and Angelina Jolie appeared in?	- TYPE TOKENS: [WDT/What, NNS/films] - TYPE PATTERN: [WHAT, NOUN] - VERB: have - VERB TOKENS: [VBP/have] - OBJECT: Brad Pitt and Angelina Jolie - OBJECT TOKENS: [NNP/Brad, NNP/Pitt, CC/and, NNP/Angelina, NNP/Jolie]

Figura 23. Pregunta compuesta y su clasificación en estructuras sintácticas.

Para la solución de este tipo de preguntas, se parte de que toda pregunta compuesta puede ser dividida en preguntas simples cuya respuesta debe ser tratada después para identificar el subgrupo que cumple una condición en concreto. En el ejemplo de la figura, la pregunta “What films have Brad Pitt and Angelina Jolie appeared in?” puede dividirse en “What films have Brad Pitt appeared in?” **AND** “What films have Angelina Jolie appeared in?”.

De este modo, se identifican 2 etapas:

1. Formación de las preguntas simples que componen la pregunta compuesta.
2. Tratamiento de las preguntas simples para que cumplan la relación que impone la compuesta.

4.1.2.1 Formación de preguntas simples

En los casos analizados, las preguntas compuestas pueden almacenarse en dos estructuras: bien en el sujeto, bien en los complementos del sujeto. Lo primero que haremos será buscar conectores lógicos en dichas estructuras y dividir los distintos recursos identificados.

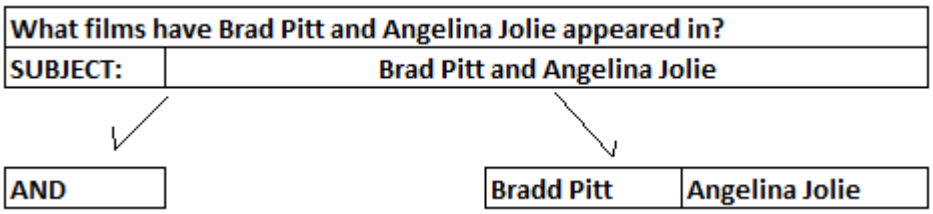

Figura 24. Descomposición de las preguntas compuestas en elementos simples.

A continuación, se formarán tantas preguntas como elementos independientes se hayan identificado. Para ello, se crearán oraciones con toda la información del resto de estructuras intacta y, en la estructura involucrada, un único elemento en cada caso.

4.1.2.2 Formación de la respuesta

Una vez se tiene el conjunto de oraciones con la información compuesta separada, las procesamos como preguntas simples, desde la fase de identificación de patrones hasta la obtención de respuesta. Una vez obtenida la respuesta de todas ellas, se debe aplicar la relación impuesta por la pregunta compuesta: en caso de ser un OR lógico, se juntarán las respuestas de todas las preguntas simples; si es un AND lógico, se buscarán los elementos coincidentes en todos los conjuntos de preguntas.

Esta implementación variará en función del tipo de respuesta de las preguntas simples; es una condición indispensable que todas ellas tengan el mismo tipo ya que no tendría sentido relacionar respuestas booleanas con conjuntos de elementos. Si las respuestas son de tipos discordantes o son respuestas textuales (simples cadenas de texto), el programa generará un error por la imposibilidad de aplicar una relación lógica entre ellas.

4.1.3 Mapeado con recursos de DBpedia

Una vez obtenida la salida de la fase de identificación del tipo de pregunta, dispondremos de los distintos recursos clasificados en función del tipo de objeto al que pertenecen. Entre los tipos conocidos, se incluyen:

- Entidades: objetos de DBpedia que se refieren a un objeto real, ya sea una persona, lugar, acontecimiento, etc.
- Propiedades: cualidad de una entidad cuyo valor puede ser otra entidad.
- Categoría: agrupación de entidades que comparten una propiedad común.

El mapeado entre el nombre del objeto en lenguaje natural y la URI que lo identifica en DBpedia, se lleva a cabo por medio de una base de datos cuyo diseño ofrece una tabla independiente para cada tipo de recurso. Se ha procurado mantener una estructura lo más abierta posible para poder hacer frente a fenómenos como la homonimia (2 palabras escritas igual con distinto significado), permitiendo relaciones N:N. A continuación se resumen las tablas empleadas para llevar a cabo los mapeos de los distintos tipos de recurso.

4.1.3.1 Mapeo de entidades

En el mapeado de entidades interviene una única tabla, llamada entityTable, que relaciona nombres propios con sus URIs.

El campo source de la tabla es en el que se almacenan los nombres de los objetos en lenguaje natural y en minúsculas (para facilitar las coincidencias, las preguntas son convertidas a minúsculas nada más entran en el sistema) y el campo entity será su homólogo en DBpedia. Así, por ejemplo, la entidad España tendrá en source el valor "spain" y entity "http://dbpedia.org/resource/Spain".

La relación entre ambos campos será unívoca (1:1) ya que de todas las páginas de DBpedia que identifican a un mismo recurso, solo nos quedaremos con la versión inglesa que además de ser la más completa, es la que corresponde al idioma que estamos procesando.

entityTable	
source	**entity**
cold war	http://dbpedia.org/resource/Cold_War
naughty dog	http://dbpedia.org/resource/Naughty_Dog
brad pitt	http://dbpedia.org/resource/Brad_Pitt
sara carbonero	http://dbpedia.org/resource/Sara_Carbonero

Figura 25. Tabla de mapeado de entidades.

4.1.3.2 Mapeo de categorías

Las categorías en DBpedia incorporan recursos de muchas páginas de Linked Data y, por ello, las URIs son de lo más variadas. Para simplificar el funcionamiento del sistema, sólo se han utilizado categorías procedentes de la propia DBpedia, es decir, aquellas cuya URI siga el patrón: dbpedia.org/resource/Category:NameOfCategory o dbpedia.org/class/yago/NameOfCategory.

La tabla empleada es equivalente a la usada para mapear entidades, almacenando por un lado el nombre en lenguaje natural de la categoría y por otro la URI que lo identifica en DBpedia. La única diferencia reside en que la relación será N:1, mapeando al mismo recurso los nombres en singular y plural que se refieran a la misma categoría, por ejemplo, "novelist" y "novelists" o "company" y "companies", cuyo uso dependerá de cómo aparezcan en la pregunta del usuario y nos permite simplificar el proceso.

typesTable	
source	**type URI**
american novelist	http://dbpedia.org/class/yago/AmericanNovelists
american novelists	http://dbpedia.org/class/yago/AmericanNovelists
company	http://dbpedia.org/ontology/Company

Figura 26. Tabla de mapeado de categorías.

4.1.3.3 Mapeo de propiedades

El mapeo de propiedades es más complejo que el de los recursos anteriores debido a que el impacto de la homonimia es mayor en este punto. Además, las propiedades pueden venir definidas por verbos (y no solo por nombres como ocurría antes), lo que hace que haya que tener una mayor flexibilidad para el procesado de los distintos tiempos verbales que puedan aparecer. Para ello, se ha generado una tabla que pretende convertir la conjugación que aparece en la pregunta analizada a infinitivo:

toInfinitive	
conjugation	**infinitive**
found	found
founded	found
were founded	found
did found	found

Figura 27. Tabla de conversión de verbos conjugados a infinitivos, con el verbo "found".

Con esto que se pretende evitar la sobrecarga producida por la homonimia de la tabla de mapeo, que aun así tendrá que mantener cierta flexibilidad para identificar la sinonimia y llevar a cabo una desambiguación de determinados términos conflictivos. Adicionalmente, se introduce una tabla que, igual que en los casos anteriores, mapea los recursos en lenguaje natural a objetos de DBpedia:

propertyTable		
source	**property**	**range**
capital	dbp:capital	dbo:place
found	dbo:foundedBy	dbo:Agent
locate	dbo:isPartOf	null

Figura 28. Tabla de mapeado de propiedades.

El infinitivo encontrado en la tabla toInfinitive (o el nombre encontrado en la oración, si la propiedad no es un verbo), se buscará sobre la tabla de propiedades, obteniendo el nombre del recurso que conoce la DBpedia de la columna "property". Adicionalmente, se guarda el rango de la propiedad, es decir, la categoría principal del recurso que aparece como valor de dicha característica (lugar, persona, etc.). En la actualidad el sistema no hace uso de dicha columna pero se ha considerado la posibilidad de identificar qué tipo de objeto se espera como respuesta en función de la pregunta (si es WHERE, esperaremos un lugar, si es WHO una persona…) y que esta información extra permita una desambiguación más precisa a la hora de realizar el mapeo. En algunos casos el rango no se puede definir unívocamente, por lo que se le da valor nulo.

4.1.4 Consultas sobre DBpedia

El módulo que gestiona las consultas sobre la base de conocimiento está dividido en dos partes:

- Submódulo de consultas básicas, encargado de la realización de consultas genéricas cada una orientada a los casos más frecuentes de extracción de información, como puede ser:
  - Valor de la propiedad de un objeto
  - Listado de categorías a las que pertenece una entidad
  - Búsqueda de entidades coincidentes con una cadena dada
- Submódulo de consultas específicas del sistema, que contiene las consultas extraídas de los diagramas de decisión de los tipos de pregunta procesados. Este

módulo no accede directamente a DBpedia sino que combina consultas básicas y procesa los resultados obtenidos de ellas para obtener la respuesta específica.

Las consultas específicas del sistema han sido confeccionadas en función de la información disponible en cada caso y la relación existente entre la información conocida y la respuesta esperada. Se distinguen las siguientes:

1. **Definición**: tipo de pregunta en la que se pide un breve resumen con la información más destacable de la entidad dada. Esta información puede contener una pequeña biografía si es una persona, una descripción si es un lugar o un acontecimiento histórico. Esta información aparece siempre contenida en la propiedad dbo:abstract del objeto.
2. **Propiedad de un objeto**: consulta en la que se pregunta por el valor de una propiedad de una entidad dada. Los recursos necesarios son el nombre de la propiedad que se desea conocer y la URI de la entidad en cuestión. El tipo de respuesta dependerá de las características de la propiedad, pudiendo ser una cadena de texto simple u otro recurso de DBpedia (como otra entidad, una categoría, etc.).
3. **Propiedad de un objeto con un tipo concreto**: variación de la pregunta anterior en la que además se pone como condición que la respuesta pertenezca a una categoría determinada.
4. **Entidad perteneciente a un determinado tipo**: se pregunta si una entidad pertenece a una categoría en concreto. La respuesta es booleana, pudiendo ser únicamente afirmativo o falso.
5. **Entidades con una determinada etiqueta**: conociendo únicamente una breve secuencia de texto, se identifican las entidades de DBpedia que contienen dicha etiqueta. La respuesta es una lista de entidades candidatas, con una puntuación en función de la probabilidad de referirse a ella (basada en el número de repeticiones de dicha entidad en la búsqueda de la etiqueta; cuantas más coincidencias, mayor probabilidad).

4.1.5 Gestión de respuesta: Comunicación Interna

El programa desarrollado está integrado por 3 partes independientes: el sistema de búsqueda de respuestas, la interfaz con el usuario y el módulo de gestión de errores. Las comunicaciones entre las distintas partes es gestionada desde la interfaz de usuario; durante la ejecución, el sistema repite el siguiente ciclo:

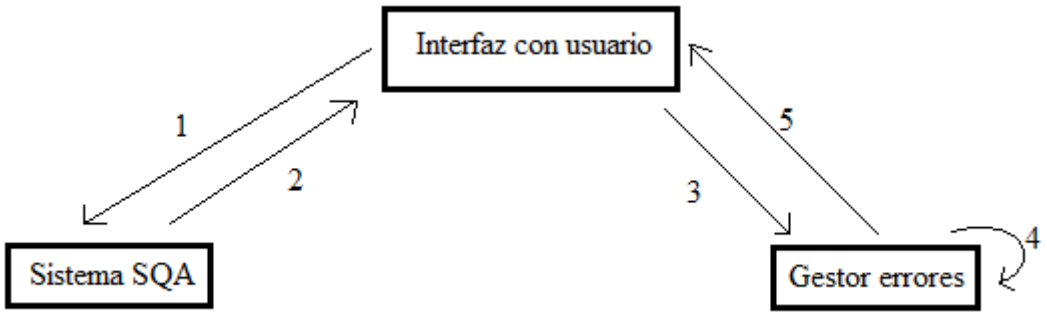

Figura 29. Interacción entre los módulos independientes del sistema.

1. El usuario introduce una pregunta en lenguaje natural al sistema. El módulo que gestiona la interacción con él, cede el control al sistema de búsqueda de respuestas.

2. El módulo SQA, procesa la pregunta introducida por el usuario, determina el patrón sintáctico que sigue y en función de ello identifica los recursos y determina la consulta que se debe realizar. Los recursos, en lenguaje natural, son mapeados a sus homólogos de DBpedia, se lanza la consulta y se obtiene la respuesta, que es devuelta al módulo anterior. Si se produce algún error a lo largo del proceso, también se notifica qué error ha tenido lugar e información adicional que pueda permitir corregirlo más adelante.
3. El módulo de interfaz con el usuario, procesa la respuesta recibida del sistema de búsqueda de respuestas y determina si es una respuesta válida o se trata de un error; en caso de ser un error, transmite la información recibida al gestor de errores para comprobar si el error se puede corregir durante la ejecución y si la respuesta es válida, la mostrará por pantalla al usuario.
4. Si el mensaje recibido era un error, la interfaz llama al gestor de errores, pasándole toda la información remitida por el módulo SQA. De todos los posibles errores del proceso de búsqueda, solo un pequeño grupo puede ser corregido sobre la marcha; cada uno de ellos tiene ligado un proceso de corrección, el cual siempre requiere la interacción con el usuario (más tarde se especificará la lista de errores corregible y el proceso a seguir).
5. Cuando el gestor de errores encuentra la solución (o determina que no es solucionable), devuelve el mando a la interfaz de usuario que le pedirá al usuario que introduzca otra pregunta. Si el error ha podido ser corregido, el usuario podrá introducir la misma cuestión que causó el error anteriormente y esta vez obtendrá una respuesta.

Como se indica anteriormente, los errores corregibles son solo un pequeño grupo de todos los problemas que pueden surgir en el proceso de búsqueda de respuestas. Entre los errores que no tienen solución se encuentran errores en el acceso a las bases de conocimiento (ya sea DBpedia o la base de datos empleada), errores en la query lanzada, que la respuesta a la consulta no sea del tipo esperado y otros de distinta índole.

Los errores corregibles son aquellos errores producidos por un error tipográfico en la introducción de la pregunta por parte del usuario o un desconocimiento por parte del sistema del recurso de DBpedia equivalente al introducido por el usuario en lenguaje natural. Por ejemplo, si en la base de datos el nombre de referencia de la Universidad Autónoma de Madrid fuera “universidad autónoma madrid” y el usuario se refiriera a ella diciendo únicamente “universidad autónoma”, la base de datos no encontraría ninguna coincidencia y devolvería un error.

La solución de estos errores pasa por buscar en DBpedia coincidencias con el nombre conocido y listar al usuario las alternativas encontradas, pidiéndole que seleccione la que se parezca más a la que hacía referencia. Dicho par nombre-recurso se guardará entonces en la base de datos para futuras menciones.

En el caso concreto de errores de mapeo de propiedades (que como hemos visto pueden ser referidas mediante verbos), la actualización de la base de datos no solo con la referencia explícita introducida por el usuario sino también con todos los sinónimos de esta. En la versión entregada en el momento de escribir esta memoria no está implementado este procedimiento pero su realización permitiría una mayor flexibilidad por aportar más información al sistema en menos interacciones: no es necesario que el usuario introduzca todos los posibles sinónimos de una propiedad para que la base de datos tenga visión de todas ellas; con que introduzca una, el resto serán añadidas automáticamente.

El procedimiento de comunicación entre los distintos módulos, ya sea para la transmisión de errores o de respuestas válidas, se lleva a cabo mediante una clase concreta que aporta información al sistema para que la interacción con el usuario sea más natural, además de hacer transparente el resultado de cada una de las fases llevadas a cabo. Además, contiene una enumeración con todos los errores conocidos y tipos de respuesta posibles (elemento de DBpedia, respuesta booleana o literal). La estructura tendrá un campo para cada posible tipo de respuesta y otro campo para información adicional de error (en el que se guardarán datos relevantes que puedan ayudar a la solución de los errores como hemos visto anteriormente).

En el caso de la interfaz, esta clase le permitirá modificar cómo devuelve la respuesta al usuario: si la respuesta es booleana, adornará la frase para decir 'yes' o 'no', si es un elemento lo contextualizará con una frase similar a la empleada por ejemplo por Siri ("esto es lo que he encontrado sobre…") y si es un literal, lo mismo. Cuando detecta que el tipo de la respuesta es un error, llamará al gestor de errores y este podrá determinar rápidamente cómo solucionarlo.

5 Integración, pruebas y resultados

Las pruebas realizadas sobre el sistema final se han hecho en un entorno controlado, debido a la ausencia (en este momento), del proceso de corrección de errores de mapeo. Para ello, se han introducido en la base de datos una serie de entidades, propiedades y categorías y se ha llevado a cabo el proceso de pruebas con preguntas que contuvieran únicamente dichos recursos. La interacción con el usuario y la variación del modo de ofrecer la respuesta del sistema en función del tipo de preguntas si ha sido testeado, como se muestra a continuación.

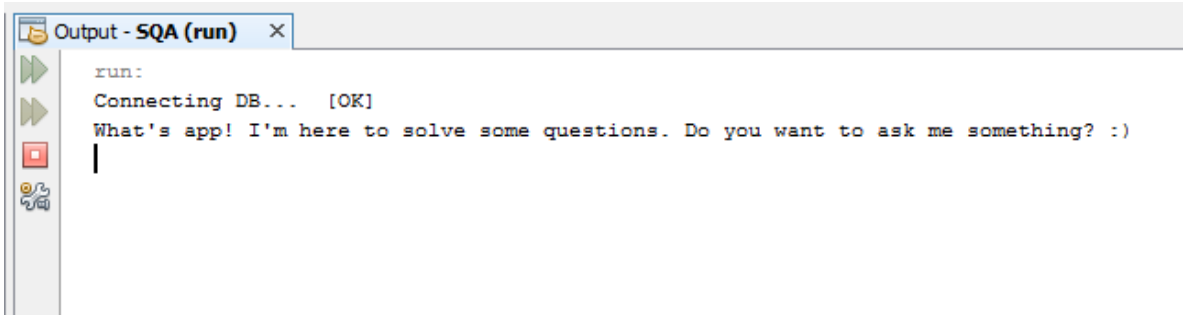

Figura 30. Bienvenida del sistema.

Se muestra por pantalla el proceso completo que lleva a cabo el sistema en el proceso de identificación de patrón y respuesta de la pregunta. A continuación, se muestra la salida del sistema ante la pregunta: "What american novelists founded Naughty Dog?", en la que idealmente, tendría que obtener todos los fundadores de dicha empresa y devolver únicamente los que pertenezcan a la categoría American Novelists.

```
[WHAT PROPERTY VERB NOUN || WHICH TYPE VERB(PROP.) NOUN]
Error while mapping
propertyOf() --- Property: dbo:foundedBy of: http://dbpedia.org/resource/Naughty_Dog with type: http://dbpedia.org/class/yago/AmericanNovelists
====
Look what I found about that:
{http://dbpedia.org/resource/Andy_Gavin=[http://dbpedia.org/resource/Category:1970_births, http://dbpedia.org/resource/Category:21st-century_Ame
```

Figura 31. Salida de la pregunta "What american novelists founded Naughty Dog?"

Se puede observar que la pregunta introducida coincide con dos patrones sintácticos: bien "What property verb noun?" (cuál es el valor de una determinada propiedad del sujeto), bien "Which type verb(prop.) noun?" (en la que la propiedad viene dada por el verbo y se especifica además que el sujeto debe pertenecer a un cierto tipo). El sistema intentará a continuación resolver la pregunta siguiendo ambos caminos; al intentarlo por el primero de ellos, falla al mapear algunos elementos y muestra "error while mapping". Entonces, prueba con el segundo camino y en este caso consigue encontrar todos los recursos que necesitaba. Realiza la consulta y con una pequeña frase introductoria, nos indica que Andy Gavin es el único fundador de Naughty Dog que cumple esa propiedad.

Otro tipo de respuesta distinto del anterior, son las respuestas booleanas, esto es, respuestas de sí o no, en las que el sistema debe comprobar únicamente la existencia de alguna propiedad o categoría. Por ejemplo, la pregunta "Is Sara Carbonero married?" buscará si el recurso de Sara Carbonero tiene la propiedad dbo:partner, que hace referencia al marido o mujer del objeto (siendo éste una persona, claro).

```
Is Sara Carbonero married?
[IS* VERB NOUN]
propertyOf() --- Property: dbo:partner of: http://dbpedia.org/resource/Sara_Carbonero
====
As far as I know, yes.
```

Figura 32. Ejemplo de pregunta booleana: "Is Sara Carbonero married?"

Al llevar a cabo la consulta, ha obtenido un resultado en la propiedad indicada y la respuesta será **true**, que el sistema "traduce" a lenguaje natural. En las preguntas compuestas, el procedimiento es equivalente.

```
Are Sara Carbonero or Brad Pitt married?
[IS* VERB NOUN]
propertyOf() --- Property: dbo:partner of: http://dbpedia.org/resource/Sara_Carbonero
[IS* VERB NOUN]
propertyOf() --- Property: dbo:partner of: http://dbpedia.org/resource/Brad_Pitt
====
As far as I know, yes.
```

Figura 33. Ejemplo de pregunta compuesta: "Are Sara Carbonero or Brad Pitt married?"

Puede verse cómo se divide la pregunta compuesta en dos simples; después, se dará respuesta a ambas y se combinará con el conector lógico AND.

Por último, si la respuesta esperada es una definición u explicación, la salida del sistema es como sigue:

```
What is the Cold War?
[WHAT IS NOUN]
definitionOf() --- Definition of: http://dbpedia.org/resource/Cold_War
====
This is what I know:
The Cold War was a state of political and military tension after World War II between powers in the Western Bloc (the United States, its NATO al
```

Figura 34. Ejemplo de definición: "What is the Cold War?" (la definición continúa pero es cortada en la imagen).

Como se ha explicado anteriormente, el procedimiento de corrección de errores se encuentra inacabado. En cualquier caso, la comunicación entre módulos sí funciona y cuando se introduce un elemento desconocido, el sistema se lo hace saber al usuario (aunque no lleve a cabo la corrección).

```
Who is Richard Stallman?
[WHO IS NOUN]
Error while mapping
I dont really know what you mean with richard stallman.Is one of the following? (insert the number aside the correct equivalent entity; if there's none, insert
```

Figura 35. Ejemplo de error al mapear la entidad Richard Stallman.

Se detecta que el recurso Richard Stallman no está mapeado en la base de datos y se muestra una lista de posibles coincidencias. Sin embargo, por el momento la lista no se muestra de manera correcta.

Las demás pruebas realizadas han consistido en dividir el proceso general en los módulos comentados en el apartado anterior y testear cada uno por separado, que se puede resumir en las siguientes etapas:

Testeo inicial de la librería NLP de Stanford para comprender su funcionamiento y determinar el procedimiento a seguir para extraer la información necesaria para el programa.

Una vez generado el módulo que ordenaba en las estructuras explicadas en el apartado 4.1.1, fase de testeo con distintas preguntas de ejemplo para ver cómo se repartía la información en las distintas estructuras para poder establecer patrones y de aquí extraer los diagramas de decisión para cada tipo de pregunta y estructura sintáctica.

Una vez programados los árboles de todos los tipos de pregunta encontrados, fase de pruebas sobre las preguntas usadas para su formación, comprobando que se identificaban los distintos recursos de manera correcta.

A partir de la lista de preguntas de origen, generación de las consultas que daban la respuesta apropiada y relación entre éstas y los diagramas de decisión del punto anterior.

Por último, pruebas de introducción de preguntas y obtención de respuestas e integración de la interfaz de usuario.

6 Conclusiones y trabajo futuro

6.1 Conclusiones

Los sistemas de búsqueda de respuesta son el futuro de los motores de búsqueda que conocemos actualmente. La flexibilidad que ofrecen y las enormes facilidades que dan al usuario para interactuar con ellos, hacen que cualquier persona, sean cuales sean sus habilidades con herramientas informáticas, puedan buscar y entender los resultados obtenidos. Tanto es así que los grandes motores de la actualidad, como Google, están empezando a adaptarse a este principio.

Figura 36. Ejemplo de búsqueda de respuestas en Google.

Sin embargo, hay una gran limitación en estos sistemas: la enorme complejidad de los lenguajes humanos hace que sea practicamente imposible abarcar todas las posibles preguntas que permite la sintaxis de un idiomay, para más inrri, la sintaxis suele ser información parcial completada por un contexto, distinto para cada usuario, que el sistema dificilmente puede conocer. Esto hace que los sistemas de búsqueda de respuestas sean un problema no resuelto, en el que las distintas implementaciones desarrolladas intentan abarcar o bien tópicos concretos o bien unas estructuras sintácticas predefinidas para hacer abarcable un problema tan complejo.

La Web Semántica es un gran aliado para este tipo de sistemas, por ofrecer una base de conocimiento de gran tamaño muy accesible; mientras los gigantes de la informática hacen uso de fuentes de información propietarias e inaccesibles, la Web Semántica ofrece una alternativa gratuita y muy completa. De alcanzar una mayor relevancia y convertirse en estándar en la web, sería la reinvención más significativa de la tecnología tal como la conocemos, ya que los sistemas de búsqueda no serían los únicos beneficiados: se daría lugar a que cualquier sistema autónomo accediera e interactuara con la información de Internet como lo haría un humano, lo que abre una enorme cantidad de posibilidades.

En cuanto al trabajo desarrollado, se considera alcanzado con un grado de satisfacion alto el objetivo de desarrollar un sistema de búsqueda de respuesta de dominio abierto y hacerlo de modo que fuera fácilmente reemplazable el idioma usado como fuente. Para ello, el diseño del sistema se ha basado en módulos, algunos de ellos dependientes del idioma, pero que podrían ser cambiados por otros análogos para otros idiomas.

Destacar que las tecnologías usadas como base para el trabajo posee un alto potencial para mejorar el sistema desarrollado. Tanto los sistemas de búsqueda de respuesta como la Web Semántica han ido creciendo en los últimos años y su crecimiento va a seguir a gran ritmo en los siguientes, cambiando posiblemente alguno de nuestros hábitos más comunes en internet.

6.2 Trabajo futuro

Como se ha mencionado al comienzo de la memoria, el sistema desarrollado es todavía un prototipo en el que todas las fases requieren revisión y sobre el que no se ha realizado una campaña de evaluaciones rigurosas y exhaustivas. Una de las primeras cosas a realizar con una versión completamente funcional, sería presentar al sistema a un concurso de Question Asnwering[9] para comprobar su eficacia.

Entre las cosas más relevantes que se han quedado pendientes, destaca el correcto funcionamiento del módulo de corrección de errores en tiempo de ejecución que, pese a estar planteado y diseñado como se ha explicado en este documento, no se ha llevado a término; la incorporación de WordNet también forma parte del trabajo a corto plazo necesario, ya que mejoraría notoriamente la búsqueda de coincidencias entre sintagmas de la entrada y elementos de la base de datos, permitiendo la búsqueda de sinónimos.

[9] http://nlp.uned.es/clef-qa/

7 Referencias

[1] Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: the view from here. *Natural language engineering*, *7*(4), 275-300.

[2] Green Jr, B. F., Wolf, A. K., Chomsky, C., & Laughery, K. (1961, May). Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference* (pp. 219-224). ACM.

[3] Radev, D. R., Qi, H., Wu, H., & Fan, W. (2002). Evaluating web-based question answering systems. Ann Arbor, 1001, 48109.

[4] Castells, P. (2003). La web semántica. Sistemas interactivos y colaborativos en la web, 195-212.

[5] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Scientific American, 284(5), 28-37.

[6] Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data-the story so far. Semantic Services, Interoperability and Web Applications: Emerging Concepts, 205-227.

[7] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). DBpedia: A nucleus for a web of open data (pp. 722-735). Springer Berlin Heidelberg.

[8] Ferrucci, D., Brown, E., Chu-Carroll, J., Fan, J., Gondek, D., Kalyanpur, A. A., ... & Schlaefer, N. (2010). Building Watson: An overview of the DeepQA project. AI magazine, 31(3), 59-79.

[9] Lopez, V., Motta, E., & Uren, V. (2006). Poweraqua: Fishing the semantic web (pp. 393-410). Springer Berlin Heidelberg.

[10] Cimiano, P., Unger, C., & McCrae, J. (2014). Ontology-based interpretation of natural language. Synthesis Lectures on Human Language Technologies, 7(2), 1-178.

[11] Lehmann, J., Furche, T., Grasso, G., Ngomo, A. C. N., Schallhart, C., Sellers, A., ... & Liu, D. (2012). Deqa: deep web extraction for question answering. In The Semantic Web–ISWC 2012 (pp. 131-147). Springer Berlin Heidelberg.

[12] Unger, C., Freitas, A., & Cimiano, P. (2014). An introduction to question answering over linked data. In Reasoning Web. Reasoning on the Web in the Big Data Era (pp. 100-140). Springer International Publishing.

[13] O'Neill, E. T., Lavoie, B. F., Bennett, R., Staples, T., Wayland, R., Payette, S., ... & Rudner, L. M. (2003). Trends in the Evolution of the Public Web, 1998-2002; The Fedora Project: An Open-source Digital Object Repository Management System; State of the Dublin Core Metadata Initiative, April 2003; Preservation Metadata; How Many People Search the ERIC Database Each Day? D-lib Magazine, 9(4), n4.

8 Anexos

A Ejemplos de preguntas resueltas

A continuación se muestra el listado de preguntas sobre las que se ha llevado el test del correcto funcionamiento del sistema. Nótese que la única limitación para la respuesta de preguntas que no se encuentren en esta lista (pero mantengan una estructura sintáctica similar) es la necesidad de tener los recursos empleados correctamente mapeados en la base de datos.

Pregunta	Patrón identificado	Consulta SPARQL	Respuesta

What nationality is Marco Reus?	Property: dbo:birthPlace of: http://dbpedia.org/resource/Marco_Reus	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Marco_Reus> dbo:birthPlace ?o .}
		}
			http://dbpedia.org/resource/Dortmund

What is the Cold War?	Definition of: http://dbpedia.org/resource/Cold_War	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Cold_War> dbo:abstract ?o .}
		}
			The Cold War was a state of political and military tension after World War II between…

Which is the capital of Canada?	Property: dbp:capital of: http://dbpedia.org/resource/Canada	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Canada> dbp:capital ?o .}
		}
			http://dbpedia.org/resource/Ottawa

What American novelists founded Naughty Dog?	Property: dbo:foundedBy of: http://dbpedia.org/resource/Naughty_Dog with type: http://dbpedia.org/class/yago/AmericanNovelists	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Naughty_Dog> dbo:foundedBy ?o .}
		}
			http://dbpedia.org/resource/Andy_Gavin

What American novelists or writers founded Naughty Dog?	Property: dbo:foundedBy of: http://dbpedia.org/resource/Naughty_Dog with type: http://dbpedia.org/class/yago/AmericanNovelists
	OR
	Property: dbo:foundedBy of: http://dbpedia.org/resource/Naughty_Dog with type: http://dbpedia.org/class/yago/AmericanComicsWriters
		SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Naughty_Dog> dbo:foundedBy ?o .}
		}
		OR
		SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Naughty_Dog> dbo:foundedBy ?o .}
		}
			http://dbpedia.org/resource/Andy_Gavin
			http://dbpedia.org/resource/Jason_Rubin

Who is the founder of Microsoft?	Property: dbo:foundedBy of: http://dbpedia.org/resource/Microsoft	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Microsoft> dbo:foundedBy ?o .}
		}
			VOID

Who wrote Macbeth?	Property: dbo:author of: http://dbpedia.org/resource/Voodoo_Macbeth	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Voodoo_Macbeth> dbo:author ?o .}
		}
			http://dbpedia.org/resource/William_Shakespeare
			http://dbpedia.org/resource/Orson_Welles

Who is Steve Jobs?	Definition of: http://dbpedia.org/resource/Steve_Jobs	SELECT ?o WHERE { {<http://dbpedia.org/resource/Steve_Jobs> dbo:abstract ?o .}	Steven Paul Jobs (/?d??bz/; February 24, 1955 – October 5, 2011) was an American businessman. He was best known as the co-founder, chairman, and chief …
Where is Collado Villalba?	Property: dbo:isPartOf of: http://dbpedia.org/resource/Collado_Villalba	SELECT ?o WHERE { {<http://dbpedia.org/resource/Collado_Villalba> dbo:isPartOf ?o .}	http://dbpedia.org/resource/Community_of_Madrid

		}	

Where did Salvador Dalí live?	Property: dbo:birthPlace of: http://dbpedia.org/resource/Salvador_Dalí	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Salvador_Dalí> dbo:birthPlace ?o .}
		}
			http://dbpedia.org/resource/Figueres
			http://dbpedia.org/resource/Catalonia
			http://dbpedia.org/resource/Spain

Where was born Steve Jobs?	Property: dbo:birthPlace of: http://dbpedia.org/resource/Steve_Jobs	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Steve_Jobs> dbo:birthPlace ?o .}
		}
			VOID

Is Sara Carbonero married?	Property: dbo:partner of: http://dbpedia.org/resource/Sara_Carbonero	SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Sara_Carbonero> dbo:partner ?o .}
		}
			True

Are Brad Pitt or Sara Carbonero married?	Property: dbo:partner of: http://dbpedia.org/resource/Brad_Pitt
	OR
	Property: dbo:partner of: http://dbpedia.org/resource/Sara_Carbonero
		SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Brad_Pitt> dbo:partner ?o .}
		}
		OR
		SELECT ?o WHERE {
		{<http://dbpedia.org/resource/Sara_Carbonero> dbo:partner ?o .}
		}
			True

Figura 37. Anexo A: Ejemplos de preguntas resueltas

Que la respuesta sea correcta o no, depende en gran medida de la información contenida en DBpedia. Existen algunas propiedades de ciertas entidades que no siguen las pautas generales; por ejemplo, al consultar por los fundadores de Microsoft, no obtenemos resultado porque la propiedad *dbo:foundedBy* no existe en dicha entidad, algo que sí ocurre en la inmensa mayoría de entidades de DBpedia.

B Diagramas de decisión implementados

En este anexo se muestran los diagramas de decisión desarrollados. Como se explica en el apartado 4.1, se utiliza como base la información contenida en las estructuras semánticas obtenidas a partir del análisis sintáctico de la oración. Véase dicho apartado para conocer cómo se forman dichas estructuras.

En las imágenes adjuntas, dichas estructuras aparecen mentadas como *tipo*, *sujeto* y *complemento*.

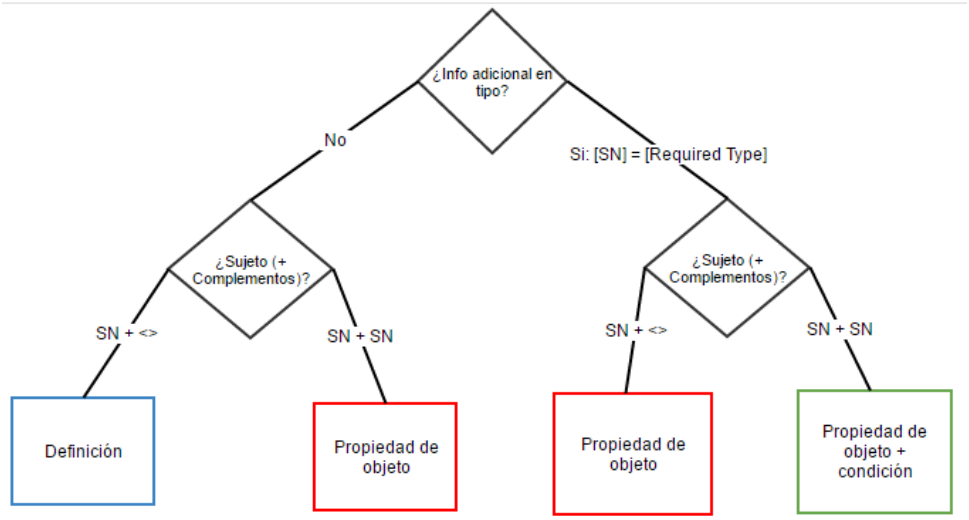

Figura 38. Anexo B: Diagrama de decisión WHAT-IS

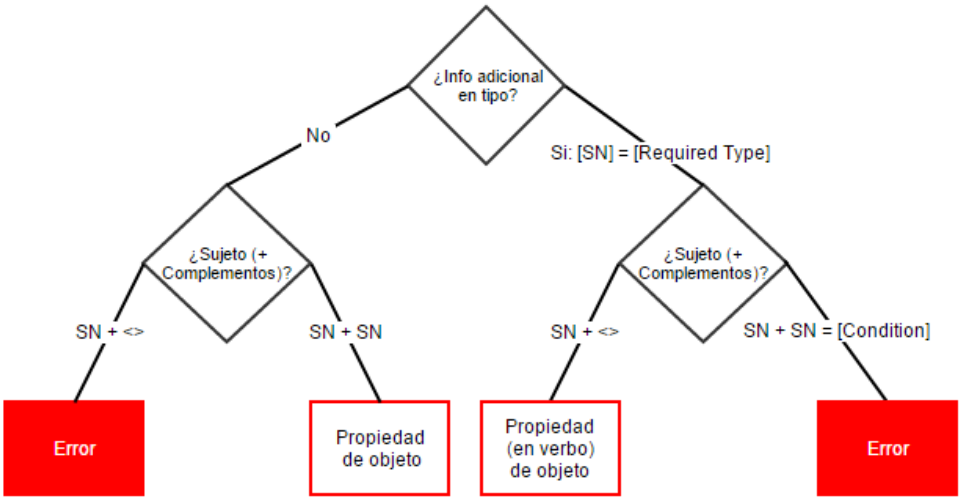

Figura 39. Anexo B: Diagrama de decisión WHAT

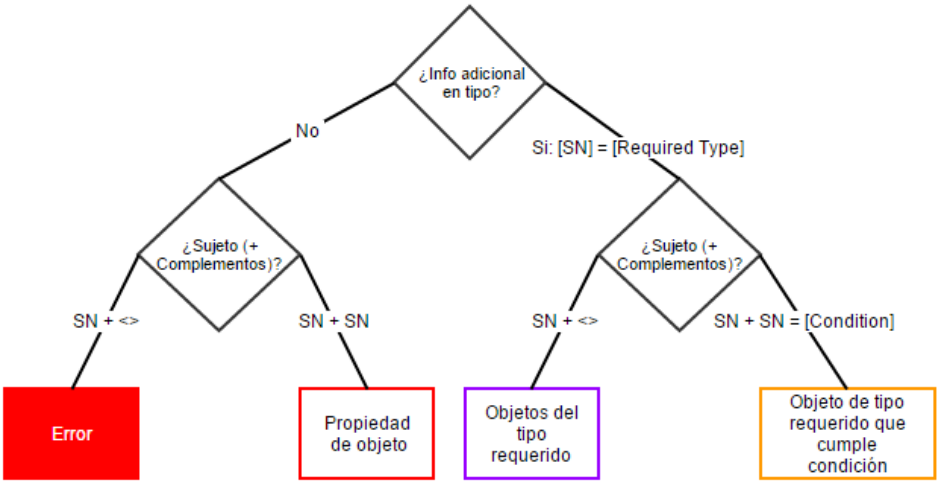

Figura 40. Anexo B: Diagrama de decisión WHICH-IS

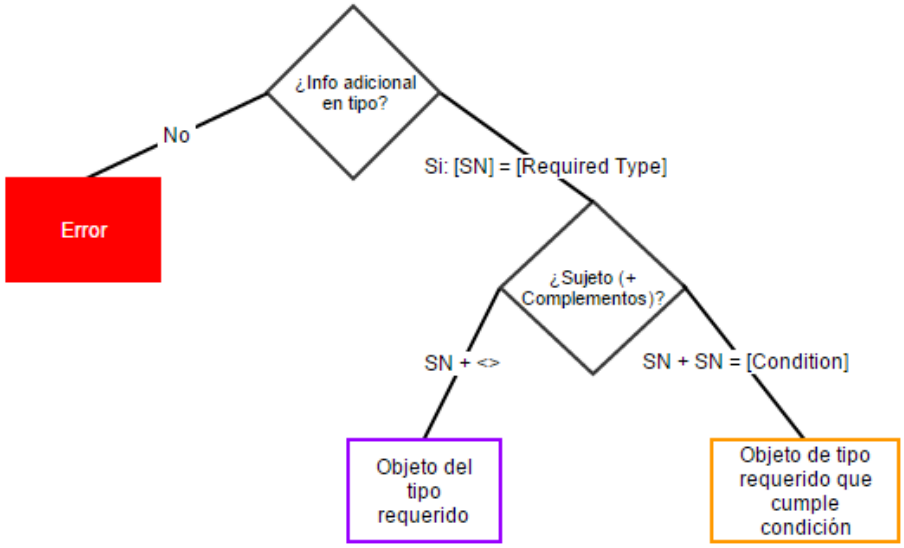

Figura 41. Anexo B: Diagrama de decisión WHICH

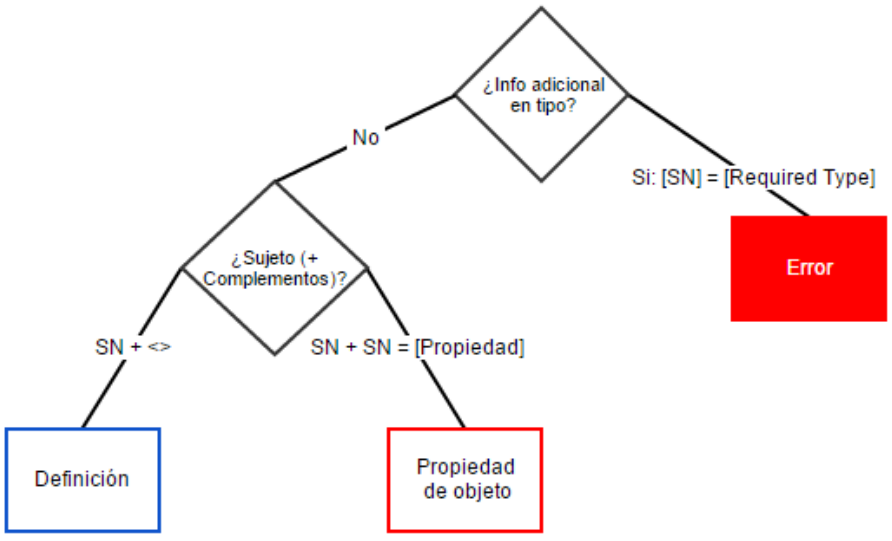

Figura 42. Anexo B: Diagrama de decisión WHO-IS

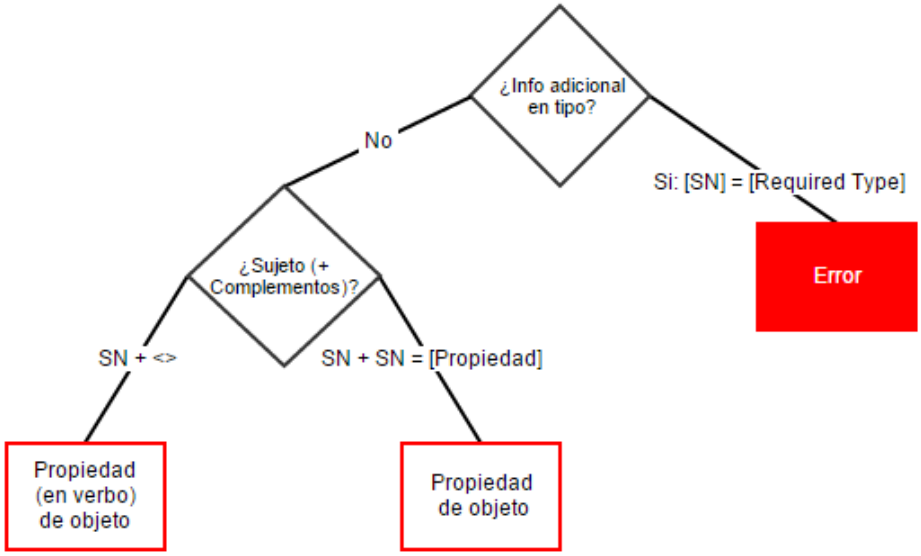

Figura 43. Anexo B: Diagrama de decisión WHO

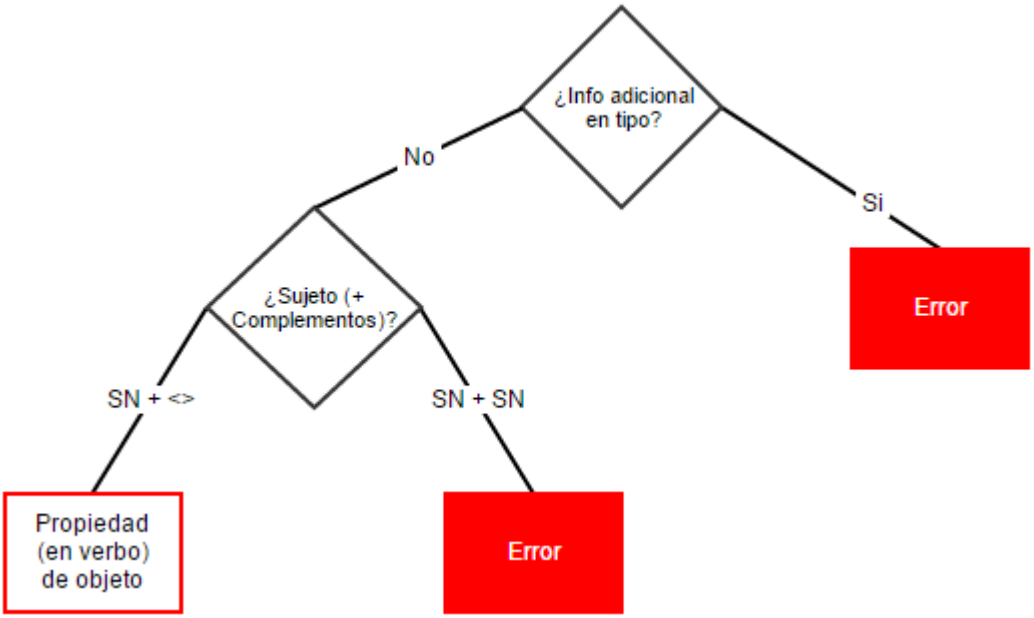

Figura 44. Anexo B: Diagrama de decisión WHERE-IS

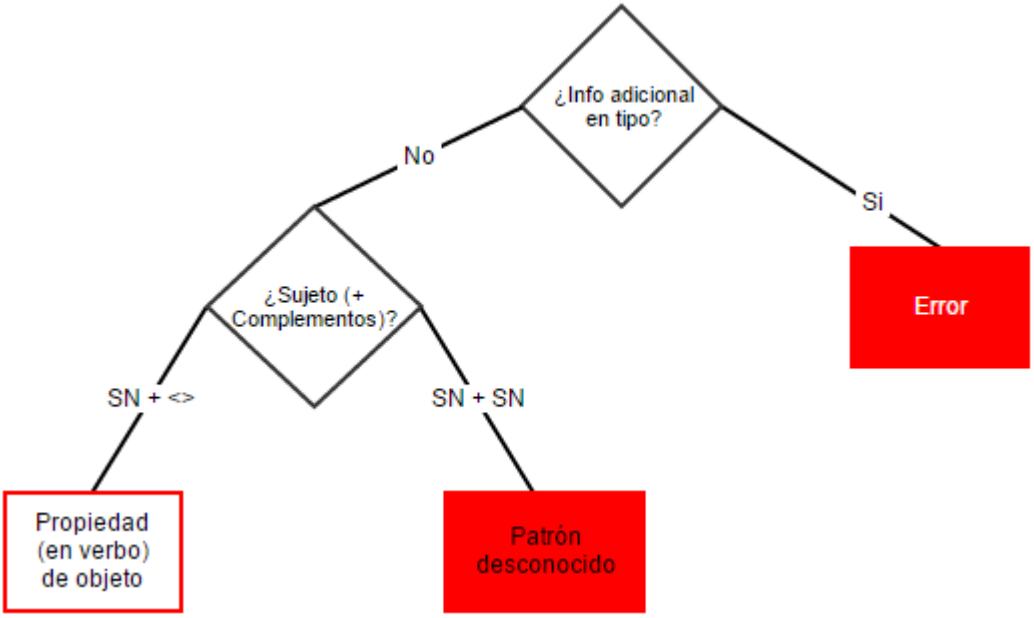

Figura 45. Anexo B: Diagrama de decisión WHERE

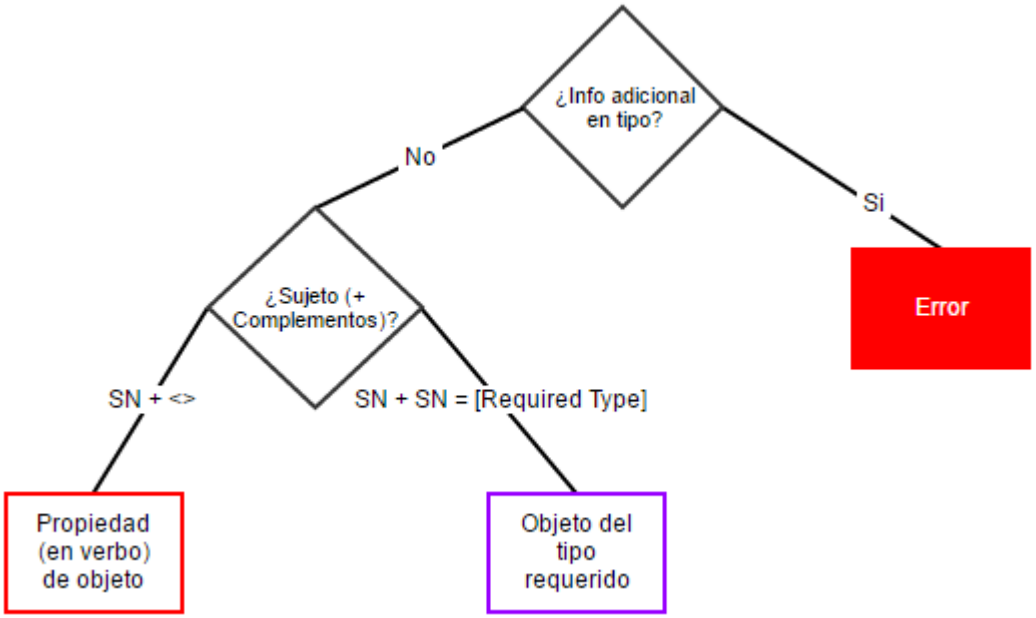

Figura 46. Anexo B: Diagrama de decisión YES-NO